Titre: Title:	Reactor Physics of a Deep-Burner Prismatic Core for VHTR
Auteur: Author:	Vincent Michel Descotes
Date:	2011
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Descotes, V. M. (2011). Reactor Physics of a Deep-Burner Prismatic Core for VHTR [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/536/

UNIVERSITÉ DE MONTRÉAL

REACTOR PHYSICS OF A DEEP-BURNER PRISMATIC CORE FOR VHTR

VINCENT MICHEL DESCOTES
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉNERGÉTIQUE)
AVRIL 2011

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

REACTOR PHYSICS OF A DEEP-BURNER PRISMATIC CORE FOR VHTR

présenté par : DESCOTES Vincent Michel
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. MARLEAU Guy, Ph.D, président.
M. HÉBERT Alain, D.Ing., membre et directeur de recherche.
M. KOCLAS Jean, Ph.D., membre et codirecteur de recherche.
M. ORTENSI Javier, Ph.D., membre et codirecteur de recherche.
M. CHAMBON Richard, Ph.D., membre.

À ma famille.

ACKNOWLEDGEMENTS

This master's thesis is the result of a collaboration with my research director and the Idaho National Laboratory. I would first like to thank my research director Alain Hébert for offering me this opportunity to study reactor physics in a pleasant and rewarding framework. I also thank Jean Koclas for having been my co-director, Richard Chambon who accepted to be part of my jury, and Guy Marleau who was always available to answer the questions I had and accepted to preside to the jury.

As stated, this work is part of a larger project pursued by the INL on high temperature reactors. I was given the opportunity to work there with Javier Ortensi and Michael Pope, and their mentoring has been a great pleasure for me. I thank both of them for their welcoming friendship in Idaho Falls, their everyday help and all the discussions we had on this project, as well as on the nuclear industry in general. They have contributed a lot to this thesis and to my integration in Idaho Falls, and I gratefully thank them for my wonderful experience in the USA. I also thank Yaqi Wang and Cristian Rabiti from the INL for having let me use their code INSTANT for this project.

Several friends were doing their studies with me at the same time in Montréal: may I thank Frédéric Simon, Cécile Carré, Maxime Guyot and more recently Cédric Laville for their support when we were all discovering DRAGON and for the good atmosphere they created between the students of the department. I am also particularly grateful to Nicolas Martin who supported me during all my time in Montréal and who has also been directly involved in the thesis by doing some SERPENT calculations for me. I also wish to thank Thibault Gosset for his friendship and support as flat mate.

My decision to come and study nuclear engineering in Montréal has been advised by Pr. Elsa Merle-Lucotte, whom I take here the occasion to thank for having taken the time to receive me in her office in Grenoble while I was thinking about my projects of study. I have also a thought for MM. Cordier, Salanave and Vandenboomgaerde who have been teaching me nuclear engineering in the École Centrale Paris and transmitting their interest for this field, MM. Martinet and Odermatt who taught me physics in a pleasant way some years ago, and M. Gandit who introduced me to research during my studies.

Finally, my family has been a constant support for me everytime and I address them all my recognition.

RÉSUMÉ

Ce mémoire de maîtrise rend compte des problèmes de simulation d'un cœur de réacteur nucléaire à très haute température. Ce cœur possède une géométrie hexagonale prismatique. Le combustible est composé d'isotopes transuraniens et est réparti le long d'un anneau. Le centre du cœur et sa périphérie contiennent des blocs de réflecteur en graphite. Cette disposition pose plusieurs problèmes au neutronicien.

Le faible pouvoir modérateur du graphite confère aux neutrons un plus grand libre parcours moyen que dans les réacteurs à eau sous pression. Les blocs de graphite ralentissent plus les neutrons que les blocs de combustible. Ce changement du spectre neutronique se fait sentir profondément à l'intérieur du combustible situé en périphérie du cœur. La thermalisation dans le réflecteur conduit à deux pics de fission caractéristiques aux interfaces combustible - réflecteur, les taux de fission augmentant du fait de la plus grande proportion de neutrons thermiques.

Cet effet spectral complique l'utilisation du schéma de calcul classique à deux niveaux : calcul en réseau infini sur un bloc pour générer des paramètres homogénéisés, suivi du calcul de cœur avec des blocs homogènes. Nous avons testé ce schéma sur un cœur dépourvu de mécanismes de contrôle (barres, absorbants) et chargé avec un combustible neuf. Un deuxième schéma a également été mis en place dans lequel les blocs voisins du réflecteur sont simulés au niveau réseau avec leur environnement (supercellules), ceci afin de générer des sections efficaces avec un spectre corrigé, plus proche de la réalité. Un calcul de référence a été effectué avec MCNP.

Il en ressort que si le schéma avec les supercellules conduit à une amélioration de la précision, le schéma à blocs simples permet d'ores et déjà de simuler le cœur avec une précision très satisfaisante à condition de garder typiquement 26 groupes d'énergie pour le calcul de cœur. Le temps de calcul prohibitif des supercellules ne permet pas à ce stade de recommander leur utilisation. Cependant, des calculs d'évolution sur une supercellule montrent que la présence du réflecteur induit de fortes variation des concentrations au sein du bloc de combustible. Ceci indique que le calcul de l'évolution des concentrations doive probablement se faire au niveau du calcul de cœur plutôt qu'en calcul réseau. L'ajout des poisons neutroniques et des barres de contrôle devrait également plaider en faveur de l'utilisation de supercellules.

ABSTRACT

The deep-burn prismatic high temperature reactor is made up of an annular core loaded with transuranic isotopes and surrounded in the center and in the periphery by reflector blocks in graphite. This disposition creates challenges for the neutronics compared to usual light water reactor calculation schemes.

The longer mean free path of neutrons in graphite affects the neutron spectrum deep inside the blocks located next to the reflector. The neutron thermalization in the graphite leads to two characteristic fission peaks at the inner and outer interfaces as a result of the increased thermal flux seen in those assemblies. Spectral changes are seen at least on half of the fuel blocks adjacent to the reflector.

This spectral effect of the reflector may prevent us from successfully using the two step scheme — lattice then core calculation — typically used for water reactors. We have been studying the core without control mechanisms to provide input for the development of a complete calculation scheme. To correct the spectrum at the lattice level, we have tried to generate cross-sections from supercell calculations at the lattice level, thus taking into account part of the graphite surrounding the blocks of interest for generating the homogenised cross-sections for the full-core calculation. This one has been done with 2 to 295 groups to assess if increasing the number of groups leads to more accurate results. A comparison with a classical single block model has been done. Both paths were compared to a reference calculation done with MCNP.

It is concluded that the agreement with MCNP is better with supercells, but that the single block model remains quite close if enough groups are kept for the core calculation. 26 groups seems to be a good compromise between time and accuracy. However, some trials with depletion have shown huge variations of the isotopic composition across a block next to the reflector. This may imply that microscopic depletion is necessary in the calculation scheme in order to obtain accurate pin-level nuclide densities. Nevertheless, for some scoping studies, block averaged values might have the desired level of accuracy.

ZUSAMMENFASSUNG

In dieser Diplomarbeit handelt es sich um die Simulationsprobleme des Kernes eines Hochtemperaturnuklearreaktors. Dieser Kern besitzt eine sechseckige prismatische Geometrie. Der aus Transuranen bestehende Kernbrennstoff ist entlang eines Ringes verteilt. Das Kernzentrum und dessen Peripherie setzen sich aus Graphitreflektorblöcken zusammen. Diese Anordnung bereitet dem Physiker viele Probleme vor.

Das geringe Bremsvermögen des Graphits vergrößert die mittlere freie Weglänge der Neutronen im Vergleich zu Druckwasserreaktoren. Die Graphitblöcke verlangsamen die Neutronen mehr als die Brennstoffblöcke. Diese Veränderung des Neutronenspektrums kann eindeutig in einer Hälfte eines Brennstoffblocks, der sich in der Kernperipherie befindet, gesehen werden. Die Abbremsung im Reflektor führt zu zwei charakteristischen Fissionhochpünkten in den Brennstoffblöcken, die an einen der beiden Reflektoren angrenzen, weil der höhere thermische Fluss die Fissionen fördert.

Diese Spektralveränderung erschwert die Benutzung eines klassischen zwei Etappen Schemas : Gitterberechnung eines Brennstoffblocks auf einem unendlichen Netz, um homogene Wirkungsquerschnitte herzustellen, dann Kernberechnung mit homogenen Blöcken. Wir haben dieses Schema auf einem Kern ohne Regelstäben, aber voll mit neuem Brennstoff, ausprobiert. Ein zweites Berechnungsschema wurde auch gebildet, in dem die Blöcke, die angrenzend an dem Reflektor sind, während der ersten Etappe mit ihren Nachbarn simuliert werden. Deshalb werden ihre homogenen Wirkungsquerschnitten mit einem Fluss hergestellt, der näher von der Realität sein sollte. Solche Berechnungen werden als „Superblöcke“ genannt. Eine MCNP-Kernberechnung wurde auch gemacht und dient als Maßstab.

Aus dieser Arbeit geht folgendes hervor : Auch wenn das Schema mit Superblöcken zu einer Verbesserung der Genauigkeit führt, ermöglicht das klassische Schema schon gute Simulationen, allerdings nur unter der Bedingung, dass genügende Energiegruppen in der Kernberechnung benutzt werden. 26 Energiegruppen ergeben typischerweise gute Ergebnisse. Die beachtliche Berechnungszeit und die Komplexität benachteiligen das Schema mit Superblöcken. Jedoch haben Entwicklungsberechnungen auf Blöcken und Superblöcken gezeigt, dass der Reflektor zu verschiedenen Konzentrationen innerhalb des Brennstoffs führt. Es mag wohl zeigen, dass die Entwicklung der Konzentrationen eher während der Kernberechnung als während der Gitterberechnung gemacht werden sollte. Die Einführung der brennbaren Reaktorgiften und der Steuerelementen sollte auch die Berechnungsschemen mit Superblöcken fördern.

КОНСПЕКТ

Эта дипломная работа представляет проблемы моделирования активной зоны реактора на высокой температуре. Эта активная зона обладает призматической геометрией. Топливо создано из трансурановых изотопов и распределено в кольце. Центр активной зоны и его периферия содержат блоки рефлектора в графите. Это положение ставит задачи специалисту в области ядерной физики.

Слабая замедляющая способность графита дарует нейтронам более свободный средний пробег, чем в реакторах с водой под давлением. Графитовые блоки замедляют больше нейтронов чем топливные блоки. Это изменение нейтронного спектра глубоко чувствуется внутри топлива, которое располагается на периферии активной зоны. Замедление нейтронов в рефлекторе ведет к двум характерным пикам ядерного деления в расстоянии между топливом и рефлектором, потому что скорость деления увеличивается благодаря большей пропорции тепловых нейтронов.

Это спектральное изменение усложняет использование классической схемы расчета на два уровня: расчет бесконечной решетки топливного блока, чтобы генерировать гомогенные эффективные сечения, и затем расчет полной активной зоны с гомогенными блоками. Эта схема была тестирована с реактором без механизмов контроля (значит без управляющих стержней и без поглотителей нейтронов) с новым топливом везде. Вторая схема была построена, в которой блоки, соседние с рефлектором, моделируются на уровень решетки с их средой («суперблоки»). Эта техника имеет целью генерировать гомогенные эффективные сечения, используя нейтронный исправленный спектр, который принимает в расчет присутствие рефлектора. Справочный расчет был осуществлен с MCNP.

Констатируем, что схема с суперблоками ведет к улучшению точности. Между тем, классическая схема в простых блоках позволяет впредь моделировать активную зону с удовлетворительной точностью при условии — хранить 26 групп энергии в расчете полного реактора. Запретительное время расчета суперблоков не говорит в пользу их использования. Между тем, расчеты эволюции на суперблоке показывают, что присутствие рефлектора заключает сильные изменения концентраций в топливный блок. Это указывает, что расчет эволюции концентраций должен будет, вероятно, делаться в расчете полного реактора скорее чем в расчете решетки. Добавление горючих вредных поглотителей нейтронов и контрольных стержней должно было бы также благоприятствовать использованию суперблоков.

CONDENSÉ EN FRANÇAIS

Parmi les projets du forum Génération IV pour les réacteurs du futur, le réacteur à très haute température (VHTR) a été choisi comme axe de développement privilégié par l'Idaho National Laboratory avec lequel ce mémoire a été préparé. Ce réacteur suscite l'intérêt de la communauté scientifique et industrielle pour ses applications en dehors de la production électrique : il permettrait d'offrir à ses clients une source de chaleur aux alentours de 1000°C, utilisable pour la production de dihydrogène par thermocraquage de la molécule d'eau, mais aussi pour certaines industries chimiques comme la synthèse d'engrais, le raffinage du pétrole, la liquéfaction des schistes bitumineux, voire en tant que chauffage d'appoint pour les industries du verre et de l'acier. L'efficacité theromdynamique de Carnot du réacteur serait également augmentée pour atteindre environ 48 % grâce à l'augmentation de la différence de température entre la source chaude et la source froide. Une utilisation combinant la fourniture de chaleur et la production d'électricité est également envisageable.

Le cœur de ce réacteur est constitué de blocs hexagonaux prismatiques. Le combustible est réparti dans un anneau avec, situés de part et d'autre, des blocs dits de réflecteur en graphite plein. Les blocs de combustible eux-mêmes sont constitués de cellules hexagonales de différents types :

- des cellules où le combustible est réparti dans une pastille cylindrique au centre d'une cellule en graphite ;
- des cellules où passent des tubes de deux diamètres différents pour le caloporteur au centre d'une cellule en graphite ;
- des cellules de graphite pleines.

Pour résister aux hautes températures, le combustible est enfermé dans des particules sphériques en céramique dites particules TRISOs, noyées dans une matrice de graphite formant la pastille. Ce dessin est commun à tous les réacteurs à haute températures. De plus, ce concept permettrait d'atteindre des taux de combustion particulièrement élevés, de l'ordre de 600 GW jours.tonne^{-1}. Cette résistance du combustible fait du VHTR un bon candidat pour brûler une partie des isotopes du plutonium et de l'américium issus de l'exploitation du parc de réacteurs à eau sous pression. Ainsi, le réacteur étudié n'utilisera que des isotopes transuraniens (Pu-Am) dans son combustible. On considère un réacteur nu, sans barres de contrôle ni poisons neutroniques, et dont le combustible est entièrement neuf.

Le flux neutronique est une grandeur égale au produit de la densité neutronique par la vitesse des neutrons. Le calcul de cette grandeur permet de prévoir la répartition de la puissance dans le cœur, sa criticité, ses réactions en cas de transitoire accidentelle, et

d'effectuer les calculs d'évolution pour suivre les concentrations de chaque isotope dans le cœur. Le calcul de flux influence donc une grande partie des choix de conception d'un réacteur, et les méthodes de calcul font l'objet d'une vérification attentive de la part des autorités de sûreté.

Le flux est gouverné par l'équation de transport des neutrons, dans laquelle interviennent les paramètres neutroniques des isotopes présents dans le cœur. Ces paramètres sont appelés sections efficaces et varient fortement en fonction de l'énergie.

L'équation de transport s'exprime comme un problème aux valeurs propres, et la valeur propre principale est appelée facteur de multiplication ou K_{eff}. Ce facteur caractérise la criticité du cœur : sous-critique lorque $K_{eff} < 1$, sur-critique lorsque $K_{eff} > 1$ et critique lorsque $K_{eff} = 1$. Il est donc particulièrement important et fourni un moyen simple de comparer des simulations entre elles. Ces comparaisons doivent néanmoins s'accompagner d'une étude du flux ou bien des taux de réaction pour véritablement attester de la précision du calcul sur tout le domaine spatial.

La résolution de l'équation de transport en considérant un cœur entier, hétérogène, complètement détaillé au niveau géométrique n'est actuellement pas faisable, du moins pas avec les capacités de calcul actuelles. C'est pourquoi un schéma à deux niveaux a été imaginé et employé avec succès pour les réacteurs à eau sous pression (REP) et pour les réacteurs CANDU. Il consiste en :

- un calcul dit réseau sur un assemblage simple avec par exemple 295 intervalles ou groupes d'énergie. La géométrie de l'assemblage est très détaillée et répétée à l'infini. Les flux obtenus permettent en fin de calcul d'homogénéiser les sections efficaces sur l'assemblage en entier et de réduire le nombre de groupes d'énergie par condensation.
- Ces sections efficaces homogénéisées sont ensuite transférées au calcul de cœur, qui est lui effectué sur la géométrie complète du cœur mais avec des blocs homogènes et peu de groupes d'énergie. On en tire un flux sur le cœur et une carte des taux de réaction.

Dans le cas du VHTR, le cœur annulaire et le graphite introduisent plusieurs difficultés pour pouvoir mettre en œuvre un schéma à deux niveaux avec succès.

Le graphite ralentit les neutrons. On parle de modération et de thermalisation du spectre neutronique. Par rapport à l'eau, le pouvoir modérateur du graphite est relativement faible, ce qui augmente le libre parcours moyen des neutrons. L'énergie des neutrons va donc passer d'un spectre très thermalisé dans le réflecteur à un spectre plus dur dans les blocs de combustible. Le changement spectral concerne uniquement les blocs voisins du réflecteur, sur environ la moitié d'un bloc. Cette profondeur de pénétration est beaucoup plus importante que dans le cas des assemblages de réacteurs à eau, et induit des problèmes de simulation. En effet, le calcul d'un bloc simple en réseau infini produit un flux dont le spectre est assez dur,

adapté à la simulation des blocs situés au centre de l'anneau de combustible. Mais ce spectre n'est pas représentatif de celui observé dans les blocs de combustible de la périphérie interne ou externe. Or l'homogénéisation des sections efficaces en vue du calcul de cœur utilise ce flux comme fonction de pondération. Les sections obtenues sont donc passablement fausses concernant les blocs voisins du réflecteur.

Pour résoudre ce problème, nous avons dans un premier temps imaginé un nouveau type de schéma de calcul dans lequel les blocs de la périphérie sont calculés au niveau réseau en prenant en compte leur environnement de graphite. Trois calculs réseau effectués sur ces supercellules - un bloc de combustible entouré de ses six voisins, dont certains appartenant au réflecteur - viennent donc remplacer le calcul de type bloc simple pour les blocs de combustible des interfaces internes et externes du cœur. Les sections efficaces des blocs internes sont toujours calculées par un calcul réseau classique sur un bloc simple.

Dans un deuxième temps, une deuxième idée a été testée : il s'agit d'augmenter le nombre de groupes gardés pour le calcul de cœur. En effet, si la discrétisation en énergie était infiniment petite, les sections efficaces microscopiques ne dépendraient pas du flux neutroniques. Au contraire, avec peu de groupes d'énergie, l'influence du flux dans le processus de condensation devient déterminante. Il semble donc logique de penser qu'un plus grand nombre de groupes dans le calcul de cœur permettrait de diminuer l'influence du flux utilisé lors de la préparation des sections efficaces homogénéisées et donc de rattraper les erreurs d'un calcul réseau qui ne tienne pas compte de l'environnement d'un bloc de combustible. Typiquement, au lieu de ne garder que 2 groupes, nous en avons gardés 26 et même 295 dans le calcul de cœur.

L'implantation dans DRAGON de ces deux schéma de calcul nous a permis de produire différents jeux de sections efficaces. Ces grandeurs ont ensuite été transférées au calcul de cœur, effectué dans le logiciel INSTANT en transport avec la méthode P_N.

Deux types de résolution ont été utilisées dans DRAGON : la méthode des caractéristiques (modules `EXCELT:` et `MCCGT:`) et la méthode des probabilités de collision avec courants d'interface (module `SYBILT:`). Le logiciel de calcul stochastique MCNP a été utilisé à toutes les étapes des calculs réseaux et cœur pour produire des solutions de référence et valider les calculs déterministes.

Il apparaît tout d'abord que `SYBILT:` ne permet pas d'obtenir des résultats satisfaisants sur de grands domaines de type supercellule, tandis que les résultats sont comparables à ceux de la méthode des caractéristiques sur un bloc simple. Le module MOC permet lui d'obtenir des résultats similaires à ceux de MCNP, mais la durée de calcul devient rapidement prohibitive, en particulier sur des domaines type supercellules où la résolution demande environ 82 heures de calcul.

L'utilisation des supercellules permet effectivement d'augmenter la précision des calculs, notamment sur les cartes de fission où l'erreur vis-à-vis de la carte produite par MCNP passe de $[-1\ \%; +1,88\ \%]$ avec le schéma « blocs simples » à ± 1 % avec le schéma « supercellules ». Cependant, les résultats sans supercellules restent généralement satisfaisant, et ceci en particulier lorsque le nombre de groupes est suffisamment important, supérieur à 12. La prédiction du facteur de multiplication est légèrement meilleure avec des supercellules, mais il est difficile de dire si cette amélioration est significative : on obtient des valeurs comprises entre ± 20 pcm dans les deux cas.

A première vue, il semble donc que l'utilisation des supercellules, si elle apporte un léger gain en précision, n'est pas justifiée compte-tenu du temps de calcul prohibitif qu'elles nécessitent. Cependant, cette conclusion n'est valable que pour un cœur dépourvu de mécanismes de contrôle et notamment de poisons neutroniques. Leur introduction pourrait complètement modifier ce point de vue. L'accélération de la méthode des caractéristiques dans DRAGON permettraient aussi de les rendre plus attractives.

Par ailleurs, un schéma de calcul complet nécessite de suivre l'évolution des matières nucléaires dans le cœur, afin d'actualiser les calculs. Le taux de combustion est le paramètre généralement utilisé pour caractériser l'usure d'un assemblage. Il représente la quantité moyenne d'énergie dissipée par l'assemblage par tonne d'éléments lourds initialement présents.

Classiquement, on effectue l'évolution sur un assemblage en calcul réseau. Les sections efficaces macroscopiques sont homogénéisées, condensées à 2 groupes et tabulées en fonction du taux de combustion de l'assemblage. Le calcul de cœur donne une carte des puissances qui, associée à un pas de temps, permet de calculer l'incrément de taux de combustion pour chaque assemblage dans le cœur. Les sections macroscopiques correspondantes sont alors récupérées dans la base de donnée, éventuellement au moyen d'une interpolation, et le calcul de cœur peut être mis à jour pour le prochain pas de temps.

Cette technique repose sur trois approximations, très bien vérifiées dans les réacteurs à eau :

- Le couplage spectral entre voisins est faible : l'approximation du calcul en réseau infini est donc bien justifiée pour simplifier la modélisation.
- Le spectre neutronique est le même dans tous les assemblages. À un taux de combustion donné, la composition isotopique d'un assemblage est donc toujours la même quelle que soit sa position dans le cœur. L'évolution dépend uniquement du niveau de la puissance d'assemblage.
- L'évolution est suffisamment homogène dans l'assemblage pour pouvoir moyenner les concentrations sur l'assemblage complet ou bien sur un maillage très grossier de cet assemblage (4 secteurs par exemple).

Dans le cas de notre réacteur, ces approximations demeurent probablement valides pour les blocs situés à l'intérieur de l'anneau de combustible. En revanche, leur validité semble beaucoup plus douteuse pour les blocs voisins du réflecteur. Nous avons donc effectué des calculs d'évolution avec DRAGON sur un bloc simple et un bloc environné, et comparé les résultats.

Il s'avère que dans le cas du bloc simple, les différences de concentration au sein du bloc après 1000 jours d'évolution à la puissance de 600 MW.tonne^{-1} sont faibles, au maximum de l'ordre de 8 % pour le ^{240}Pu. On observe au passage que le réacteur permet de brûler près de 95 % du ^{239}Pu initialement présent, ce qui correspond bien à une des missions de ce réacteur.

Dans le cas d'une supercellule, les différences sont beaucoup plus importantes au sein du bloc selon qu'on se trouve proche ou non du réflecteur. Ainsi la concentration en ^{238}Pu, au lieu d'augmenter de moitié par rapport à l'état initial, a diminué de 7,3 %. La production de ^{241}Pu est de 25 points plus faible, tandis que celle en ^{242}Pu est 25 points plus haute. Les différences sont donc notables, et elles s'expliquent par la différence d'amplitude du flux, ainsi que par la différence de spectre énergétique.

Dans l'hypothèse d'un schéma classique où les calculs d'évolution sont effectués hors cœur, sur un bloc simple, il semble qu'homogénéiser les blocs pérophériques ne rendra pas compte de l'hétérogénéité de l'assemblage. On peut envisager de discrétiser assez finement ces blocs dans le calcul de cœur, calculer le taux de combustion de chaque petite région et leur attribuer des sections efficaces macroscopiques prise dans la base de donnée réacteur, différentes en fonction du taux de combustion. Cependant, cette pratique suppose qu'à un taux de combustion donné corresponde toujours une composition donnée. Cette hypothèse nous semble fragile.

Un schéma de calcul alternatif a d'ores et déjà été proposé pour résoudre une partie du problème. Une section efficace macroscopique d'un isotope est le produit de deux termes : sa section efficace microscopique et sa concentration. On peut imaginer effectuer les calculs d'évolution dans le calcul de cœur pour les concentrations, et n'aller chercher dans la base de donnée réacteur que les sections efficaces microscopiques associées au bon taux de combustion. Cette méthode, appelée micro-évolution du cœur, permettrait au moins un calcul précis de l'évolution des concentrations. Certains calculs de REP y font déjà appel.

Cependant, la question reste posée pour le calcul des sections efficaces microscopiques. En effet, elles seraient issues d'un calcul réseau souffrant d'approximations sur les concentrations, et bien sûr des mêmes problèmes que précédemment pour les changements spectraux. Une solution pourrait être de transmettre les concentrations calculées dans le calcul de cœur au calcul réseau, afin de s'assurer au moins une certaine cohérence entre les deux. La machinerie à mettre en place est alors délicate, avec beaucoup de transferts de données. Un tel système

plaiderait pour le développement d'un code de calcul réseau et d'un code de calcul de cœur dans le même logiciel, avec par exemple des maillages fins et grossiers superposés.

Une dernière solution a parfois été évoquée. Elle consiste à imposer dans le calcul réseau les courants entrants et sortants aux frontières du bloc. Ces courants seraient récupérés à partir du calcul de cœur, avec un algorithme itératif entre les deux pour converger. On aurait ainsi la possibilité de faire évoluer chaque bloc séparément dans le calcul réseau en imposant le flux du calcul de cœur, ce qui résoudrait du même coup le problème de l'évolution et de la génération de sections efficaces avec un spectre adapté à chaque assemblage pour le calcul de cœur. La faisabilité et l'opportunité d'un tel calcul sous contraintes font néanmoins l'objet de débats entre les laboratoires. Ceci implique en effet l'abandon du modèle du mode fondamental, actuellement à la base de tous les calculs de réseau. Des recherches théoriques sont nécessaires pour clarifier le sujet.

En conclusion, il est apparu avec ce travail de recherche que la prise en compte du réflecteur lors de la génération des sections efficaces pour le calcul de cœur n'est pas véritablement nécessaire *pour calculer un cœur nu, neuf, sans mécanismes de contrôle ni poisons neutroniques*. Garder 26 groupes dans le calcul de cœur permet de pallier les approximations du calcul réseau sur un bloc simple. En revanche, l'ajout des barres de contrôle et surtout des poisons neutroniques devrait rendre la prise en compte de l'environnement inévitable durant la génération des sections efficaces pour le calcul de cœur. En effet, d'une part les poisons se situent pour certains dans la zone d'influence du réflecteur où les changements spectraux sont importants, et d'autre part l'atténuation du flux due à la présence des poisons se fait sentir sur environ une dizaine de centimètres autour de chaque poison neutronique, ce qui affecte les voisins du bloc considéré puisque les poisons se situent à la périphérie des blocs. L'utilisation des supercellules permet d'ores et déjà de diminuer significativement l'erreur observée sur les cartes de fission.

Enfin, il a été montré combien les concentrations dans les blocs périphériques varient selon la distance au réflecteur. Cette étude a aussi permis d'observer l'importance des changements spectraux à travers les blocs jouxtant le réflecteur. Homogénéiser ces blocs ne semble donc pas représentatif de la réalité, et nous suggérons au minimum de discretiser assez finement ces régions dans le calcul de cœur pour leur assigner des sections efficaces correspondant à leur taux de combustion. Il nous semble également judicieux d'implémenter dans le calcul de cœur un module de micro-évolution pour calculer les concentrations isotopiques à travers le bloc, les sections efficaces microscopiques étant cependant toujours calculées au niveau réseau. Un calcul d'évolution classique sur un bloc simple où concentrations et sections efficaces seraient transférées au calcul de cœur en fonction du taux de combustion semble a priori peu adapté à ce réacteur, même si ceci reste à prouver.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS AND ABBREVIATIONS

DOE	Department of Energy
INL	Idaho National Laboratory
LANL	Los Alamos National Laboratory
EPM	École Polytechnique de Montréal
DRAGON	Complete lattice code developed at the "École Polytechnique de Montréal".
INSTANT	Full-core P_N calculation code developed at the INL.
MCNP	Monte-Carlo reference code developed at the LANL.
SERPENT	Monte-Carlo lattice code developed at VTT (Finland) by Jaakko Leppänen.
VHTR	Very high temperature reactor
MOC	Method of characteristics. Solution method of the transport equation.
CP	Collision probability. Solution method of the transport equation.
P_N	Solution method of the transport equation using a polynomial expansion of the streaming operator.
S_N	Solution method of the transport equation using an angular discretization of the streaming operator. Also known as the discrete ordinates method.
`EXCELT:`	Tracking module in DRAGON. Used for the method of characteristics.
`MCCGT:`	Tracking module in DRAGON. Used for the method of characteristics.
`SYBILT:`	Tracking module in DRAGON using the collision probability method.
SYB	Abbreviation for `SYBILT:`.
DH295	"Double heterogeneity", 295 groups. Describes one calculation path where the geometry is completely described with TRISO particles.
HOM295	"Homogeneous", 295 groups. Calculation path in 295 groups starting with homogeneous cross-sections for the different cells (fuel, graphite, coolant).
HOM26	"Homogeneous", 26 groups. Calculation path in 26 groups starting with homogeneous cross-sections for the different cells (fuel, graphite, coolant).
SB	Single block model.
SC	Supercell model.
SPH	Super homogenisation.

CHAPTER 1

INTRODUCTION: DESCRIPTION OF A DEEP-BURN PRISMATIC VHTR

1.1 Interest of this reactor

Very High Temperature Reactors (VHTRs) are part of the six reactor concepts retained by the Generation IV forum to be investigated in order to prepare the next generation of nuclear plants towards 2040. By increasing the coolant temperature to 950°C, a VHTR could provide heat for industrial applications such as hydrogen production, fertilizer production, or refining industries. The U.S. Department of Energy has launched in 2005 a program named "Next generation nuclear plant" (NGNP) to build a demonstration of a VHTR by 2017. This program has been driven by the Idaho National Laboratory (INL) among other partners. This master's thesis has been done in collaboration with this laboratory and specifically under the supervision of Michael A. Pope and Javier Ortensi.

The first studies of this kind of reactor has been done by the Atomic Energy Research Establishment, Harwell, in 1956. The Harwell concept led to the building of the DRAGON reactor at Winfrith (United Kingdom) and to the Peach Bottom Reactor (USA) designed by General Atomics. These reactors reached full power in April 1966 and May 1967, respectively. A larger version of the reactor using the prismatic core design of General Atomics was operated in Fort S^{t} Vrain (USA) from 1977 to 1992.

The reactor we are considering is made up of a prismatic geometry with an annular core surrounded by blocks of graphite reflectors. This design has often been investigated for VHTRs because it enables both efficient cooling of the assemblies and upholding of the fission reactions. Contrary to the competing design of pebble-bed reactors, the prismatic geometry offers the advantage of knowing exactly the position of the fissile elements. Refuelling methods stay close to the actual industrial experience, since blocks are moved in other positions or removed according to the refuelling plan during one campaign. In addition, and unlike classical designs which utilize enriched uranium fuels, our reactor is burning transuranic elements coming from light water reactor spent fuel. This characteristic permits the reduction of radioactive wastes and their activity while burning the fissile elements which they contain and generating valuable power or heat for various industrial applications. The project was known as "Deep burn VHTR" in the laboratory, and it aimed to complete the studies done for the NGNP program which was considering a more conventional fuel based on uranium.

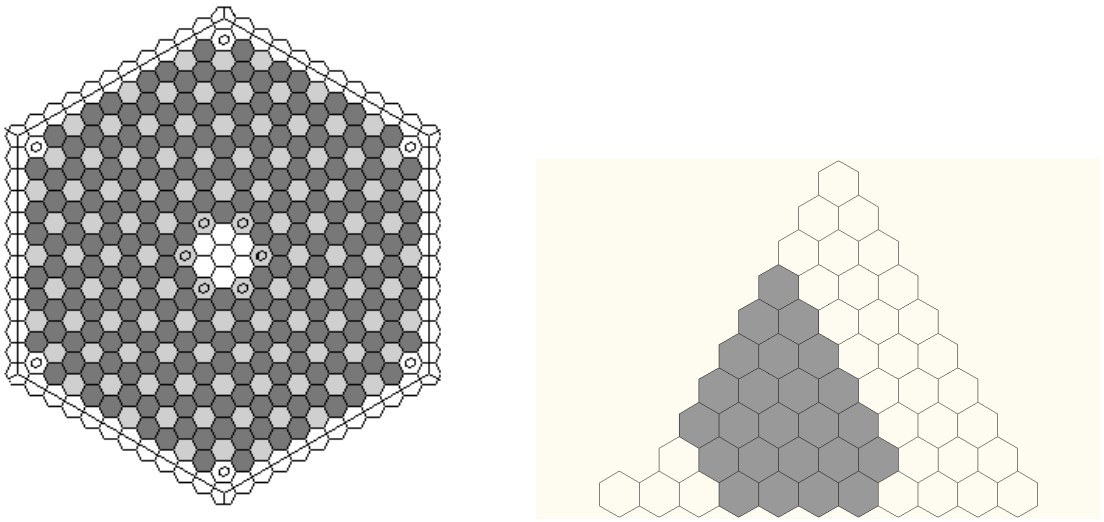
1.2 Simplified design used for this study

This thesis will focus on the neutronic simulation of this VHTR. We consider a simplified reactor with only blocks of reflector and blocks of fuel. The fuel blocks are made up of hexagonal cells of three types:

- fuel compacts surrounded by graphite and filled with transuranic elements. The fuel is located inside TRISO particles diluted in a graphite matrix;
- cylindrical tubes of coolant of two different diameters filled with helium and surrounded by graphite;
- pure graphite hexagonal cells in the center and at the periphery.

The usual places for burnable poisons in the corners of the fuel block are also filled with pure graphite. A drawing is provided by figure 1.1(a).

Figure 1.1(b) shows the 2D core map, with a central reflector block, two rings of inner reflector, then five rings of fuel blocks and three rings of outer reflector blocks before the core barrel. The core barrel's influence is not considered in our analysis as its influence on the neutronics should be low compared to the outer graphite. Therefore, we focus our study on the interaction between the fuel and the graphite reflector. In the same way, we assume that all fuel blocks are equivalent, made up of fresh fuel, without burnable poisons nor control rods.



(a) Geometry of a fuel block (b) Core map for INSTANT model

Figure 1.1 – Core map and detailed geometry of a fuel block

Table 1.1 gives the parameters used for all our calculations.

Table 1.1 – Core parameters

Parameter	Value
Block pitch (cm)	36,0
Block side (cm)	20,78461
Maximum number of blocks in a direction (N)	21
Total number of blocks (T)	331

The following relation, given in the INSTANT manual [1] gives the correspondence between the number of hexagons in a direction and the total number of hexagons in the core for a 2D complete geometry:

$$T = \frac{3}{4}(N^2 - 1) + 1 \tag{1.1}$$

The fuel blocks exhibit a $1/12^{\text{th}}$ symmetry, so that the model can be simplified in the simulation. Figure 1.2 shows the disposition of the different cells in the $1/12^{\text{th}}$ model and table 1.2(a) gives the geometric parameters. The reflector blocks are filled with graphite whose density is the same as the graphite cells in the fuel block: $8,774.10^{-2}.10^{24}$ atoms.cm^{-3} (see table 1.4). Tables 1.3(a) and 1.3(b) give the composition of the TRISO particles, and table 1.2(b) their dimensions.

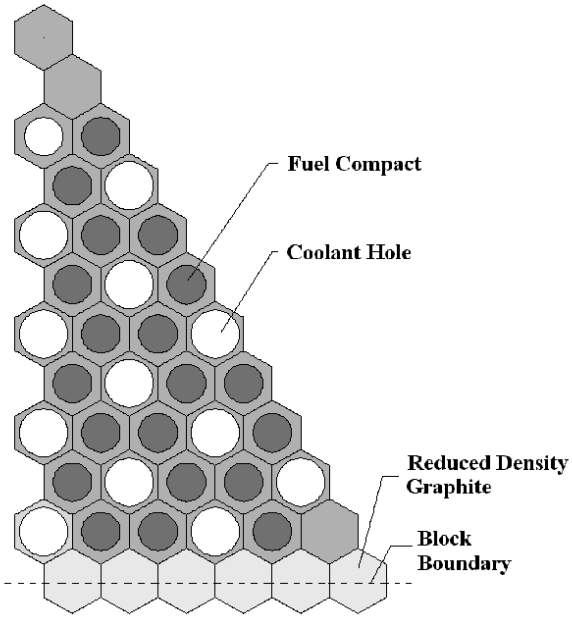


Figure 1.2 – Fuel block with $1/12^{\text{th}}$ symmetry

Table 1.2 – Fuel block parameters

(a) Cell dimensions

Parameter	Value
Large coolant channel radius (cm)	0,794
Small coolant channel radius (cm)	0,635
Fuel compact radius (cm)	0,6225
Fuel hole radius (cm)	0,6350
Coolant/Fuel cell pitch (cm)	1,88
Coolant/Fuel cell side (cm)	1,08542
TRISO packing fraction (%)	17,5016

(b) TRISO particle layer dimensions

Layer	Thickness (μm)
Fuel Kernel	200
Buffer layer	120
IPyC	40
SiC	35
OPyC	40

Table 1.3 – TRISO composition

(a) Fuel kernel

Nuclide	Number density ($.10^{24}$ atoms.cm^{-3})
^{16}O	$4{,}436.10^{-2}$
^{237}Np	$1{,}522.10^{-3}$
^{238}Pu	$6{,}464.10^{-4}$
^{239}Pu	$1{,}099.10^{-2}$
^{240}Pu	$5{,}084.10^{-3}$
^{241}Pu	$1{,}937.10^{-3}$
^{242}Pu	$1{,}074.10^{-3}$
^{241}Am	$6{,}163.10^{-4}$
^{242m}Am	$4{,}385.10^{-6}$
^{243}Am	$3{,}057.10^{-4}$

(b) Buffers compositions

Nuclide	Number density ($.10^{24}$ atoms.cm^{-3})
Porous graphite buffer layer	
Graphite	$5{,}265.10^{-2}$
IPyC layer	
Graphite	$9{,}526.10^{-2}$
SiC coating layer	
^{28}Si	$4{,}402.10^{-2}$
^{29}Si	$2{,}235.10^{-3}$
^{30}Si	$1{,}473.10^{-3}$
^{12}C	$4{,}772.10^{-2}$
OPyC layer	
Graphite	$9{,}526.10^{-2}$

Table 1.4 – Other mixtures

Nuclide	Number density ($.10^{24}$ atoms.cm^{-3})
Graphite matrix in the fuel pellet for TRISOs	$8{,}524.10^{-2}$
Graphite surrounding the fuel pellet and the coolant tubes	$8{,}774.10^{-2}$
^{4}He in the coolant tubes and in an annular tube around the fuel pellet	$7{,}000.10^{-4}$

1.3 Objective of the thesis

All reactors require numerical methods to simulate their neutronic behaviour because the equations involved are far too complicated to be analytically solved. In addition, a reactor exhibits several levels of physical scale, from the dimension of a small fuel pellet to the core dimensions, with many different media and interfaces. Moreover, the neutronic parameters of the nuclides are highly dependent on the neutron energies, which vary on a wide interval from 1 meV to 10 MeV.

The approach chosen to deal with those issues consists of two principal methods widely used for the study of light water reactors (LWRs):

– the discretization of the energy variable into small intervals called groups;

– a two-steps calculation: lattice and full-core calculation.

The lattice calculation serves to do a spatially very detailed calculation on a small domain with many energy groups, and then produce homogenized cross-sections condensed to few energy groups for the full-core calculation so that a coarser spatial mesh can be used. The lattice calculation is typically performed over a fuel block. A homogenization of the neutronic properties over some regions is performed, and then a condensation of the energy groups. Most of the current neutronic codes are now using this two-step approach.

Two-level calculations presented in this work will be performed using two codes: DRAGON Version4 and INSTANT. DRAGON Version4 is a lattice code developed at the École Polytechnique de Montréal by the institute of nuclear engineering (see references [2, 3]). INSTANT provides a full-core 3D solution based on the spherical harmonics (P_N) method. It was developed at INL by Yaqi Wang and Cristian Rabiti (see [4]). Notice that INSTANT is not a diffusion code but a transport code designed for whole core calculations.

Modelling the VHTR requires, in addition, the ability to treat in the code the hexagonal geometry, the presence of TRISO particles whose distribution in the fuel compacts are stochastic, and large lattice domains. Nowadays, DRAGON holds all those features: the hexagonal geometry is in place, the large domains may be treated by the method of characteristics or collision probability, and the double heterogeneity treatment can be used for TRISO particles.

As shown in reference paper [5], "the neutron mean free paths are significantly larger than those of LWRs." Thus the region of influence of the reflector is extended compared to LWRs. The resulting coupling effects might prevent us to successfully use the classical two-step approach of neutronic calculation, meaning a lattice transport calculation on one block, a condensation at two groups and a 2 groups diffusion calculation on the whole core. Moreover, the core has an annular design, so that the neutron distribution will be distorted:

the inner and outer blocks close to the reflector have large thermal flux peaks. This may prevent us from considering each block to exist in infinite medium and thus be represented by small-domain reflected boundary condition lattice calculations. Thus we need some means of capturing the correct neutron spectrum in reactors with large degrees of spectral penetration. One mean of achieving this is with larger domain lattice calculation. The reference paper [5] has already shown that a decoupled lattice calculation may almost be reached by considering a supercell with the two neighbouring rows of blocks next to the block of interest. In this case, we can consider the infinite lattice approximation as completely valid.

In addition we suppose that keeping a high number of groups for the whole core calculation may compensate the partial errors coming from the lattice calculation. This supposition is driven by the fact that the more discretized is the energy variable, the less influence has the flux used to homogenize the microscopic cross-sections.

In this thesis, we will investigate different ways to prepare cross-sections for the core calculation. We will especially focus on two parameters:

- the number of groups to which we condense the cross-sections before the whole core calculation.
- the presence or absence of the blocks surrounding the block of interest at the lattice level.

1.4 Organisation of the thesis report

In this thesis, neutronics equations are first presented in Chapter 2. Nothing new has been added in this chapter compared to text books, but it may be useful for the reader to review the physical equations underlying each numerical method used in the simulations. Chapter 3 will develop with details the different calculation schemes considered: single block paths versus supercells paths using a larger domain at the lattice level. Details are given about the options used in DRAGON and INSTANT. Next, Chapter 4 contains all the results to analyse the two main paths and quantify their accuracy. At the end, a depletion study is presented to provide more data for a future complete calculation scheme. It will emphasize the problems raised by this reactor. A final conclusion will summarize the results of this work and suggest some points which should be further studied. Bibliography and appendices are collected at the end of this thesis.

CHAPTER 2

NEUTRONICS EQUATIONS

The neutronics deals with interactions between neutrons and matter in a nuclear reactor. This discipline may also be called reactor physics, and is of high importance for reactor design, operation and safety. It aims to determine an accurate representation of the neutron distribution inside the core, which leads to the determination of the power shape in the reactor, the reactivity of the core, and the depletion of the different isotopes. This chapter will present the equations which underlie all neutronics codes, and particularly the codes DRAGON and INSTANT which are used in this work. All of this chapter is based on the graduate-level course of Alain Hébert which we followed in 2009 at the École Polytechnique de Montréal. An exhaustive presentation of this theme can be found in reference [6], which inspired all this chapter, but also in [7] and [8], this last one being more focused on high temperature reactors.

Interactions between neutrons and nuclei are described by functions of neutron energy and nuclide characteristics, which are called cross-sections. Their establishment results both from quantum mechanical models of the neutron-nuclei interactions and from direct or indirect measurement during experiments. Thus, neutronics only deals with the statistical behaviour of neutrons at a mesoscopic level, so that quantum effects vanish. Five assumptions – well verified – are considered [6]:

- Relativistic effects are neglected.
- Neutron-neutron interactions are neglected: indeed, the neutron density in the reactor is always much lower than the density of the nuclei.
- Neutrons are neutral particles, and therefore neutron travel in straight lines.
- The materials are isotropic in space: all considered mixtures are perfectly mixed.
- The nuclides are in thermal equilibrium with their neighbours.

The neutron distribution in the core depends on the position in the reactor (3 variables), the velocity of neutrons (3 variables), and the time (1 variable). This distribution is the solution of the *transport equation* which models the behaviour of neutrons over the core.

2.1 Fundamental nuclear parameters

Depending on the energy of the incident neutron, a collision between a neutron and a nucleus can produce several reactions. Two basic phenomena can happen:

- the neutron is scattered by the nucleus without penetrating it: it is called a potential scattering reaction. This reaction is described as elastic because both the momentum and kinetic energy of the neutron-nucleus pair are conserved. A classical analogy is the billiard-ball collision.
- the collision produces a compound nucleus where the incident neutron penetrates the nucleus and mixes with other nucleons. The compound nucleus therefore gains a lot of internal energy and is most of the time highly unstable. After a life-time of between 10^{-22} s to 10^{-14} s, the compound nucleus looses its excitation energy by emitting particles and/or electromagnetic radiations. The compound nucleus can undergo several evolutions: fission, particle emission (proton, neutron, alpha particle) and/or gamma rays emission. Notice that if a single neutron is emitted, the reaction is called *resonant scattering reaction*. The result is comparable to the potential scattering reaction. However, this reaction can be inelastic, which means that the momentum and kinetic energy are not conserved. In this case, the reaction comes with gamma ray emission. If the compound nucleus is only emitting gamma rays, then the reaction is called radiative capture [6].

In neutronics, the physicist is only interested in the result of a nuclear reaction upon the neutron population: emission, absorption or scattering of neutrons, and the balance of the reaction. The detailed knowledge of the mechanism of the nuclear reactions does not matter. Therefore, cross-sections and collision laws are introduced.

Cross-sections are related to the probability that each nuclear reaction has to occur. They lead to the calculation of corresponding reaction rates, that is to say the number of nuclear reactions of this type per unit of time.

The collision laws describe the dynamics of a collision, which leads to the calculation of the velocity and direction characteristics of the emitted particle. All reactions involving the formation of a compound nucleus are usually meant to be isotropic in the laboratory reference system (LAB). Collision laws for potential scattering reactions are usually simplified by using the approximation of isotropic scattering or of linear anisotropic scattering. Those approximations will be detailed while introducing the neutron scattering source term.

Definition of the cross-sections

Cross-sections describe the probability of each type of nuclear reaction and are based on a fundamental property of nuclear reactions [6]:

> The probability for a neutron located at $\vec{r}$ and moving in a material at velocity V_n to undergo a nuclear reaction in a differential element of trajectory ds is independent of the past history of the neutron and is proportional to ds.

For example, let us consider a mono-kinetic and monodirectional beam of neutron hitting perpendicularly a target of width ds and area S at a velocity V_n. We call I the intensity of the beam, that is to say the number of neutrons hitting the target per unit of area per unit of time. Mathematically, this intensity is defined by:

$$I = n\,V_R \tag{2.1}$$

where V_R is the relative velocity of the neutrons with respect to the target and n is the volumetric concentration of the neutrons (neutrons.cm^{-3}) in the beam. In order to simplify, let us consider the target nuclei at rest (0 K); then V_R is equal to V_n.

The volumetric concentration of nuclei in the target is given by:

$$N = \frac{\rho A_0}{M} \tag{2.2}$$

where ρ is the mass density of nuclides (g.cm^{-3}), M is the atomic mass of a nuclide (g.mol^{-1}) and A_0 is the Avogadro number defined as $6.022094.10^{23}$atoms.mol^{-1}.

We define the surface reaction rate dR_x as the number of nuclear reactions of type x per unit of time and per unit of area of the target. Experiment has shown that this reaction rate can be expressed by:

$$\mathrm{d}R_x = \sigma_x N I\,\mathrm{d}s \tag{2.3}$$

The *microscopic cross-section* σ_x is a proportionality factor. Quantum mechanics is now able to provide a more complete definition of the microscopic cross-section so that it is possible to explain part of their behaviour [9], however this is not the purpose of this thesis.

A dimensional analysis of equation 2.3 shows that the microscopic cross-section must have the dimension of an area. It is generally expressed in barns (b), with $1b = 10^{-24}$ cm^2. In a classical analogy, we can imagine the microscopic cross-section as the surface on which a neutron heading toward the nuclide will interact with it. Of course, the concept of area at those scales is not really relevant and can be misleading, but it gives a helpful representation of the phenomenon. Some nuclides will have a high microscopic cross-section, which reflects a high affinity of the neutrons with those nuclides. On the contrary, other nuclides will not react as readily with neutrons, and therefore have a small microscopic cross-section.

It can be useful to group all the characteristics of the target into a single value. Therefore we define the *macroscopic cross-section* Σ_x:

$$\Sigma_x = N\sigma_x \tag{2.4}$$

Due to its definition, the macroscopic cross-section has the dimension of the inverse of a length. It is usually expressed in cm^{-1}.

If the material of the target is a homogeneous mixture of different types i of nuclides, the resulting macroscopic cross-section is:

$$\Sigma_x = \sum_i N_i \sigma_{x,i} \tag{2.5}$$

Nuclear reactions are independent from each other. This allows us to define the *total macroscopic cross-section* as the sum of the cross-sections from all nuclear reactions:

$$\Sigma = \sum_x \Sigma_x \tag{2.6}$$

The *mean free path* λ of neutrons is defined by the average distance that the neutrons can travel in an infinite and homogeneous material before undergoing a reaction [7]. In an infinite homogeneous slab, one can demonstrate that it is equal to:

$$\lambda = \frac{1}{\Sigma} \tag{2.7}$$

where Σ is the total macroscopic cross-section [6]. The mean free path gives a rough estimate of the travelling distance of the neutrons.

Nuclear reactions are usually classified into two categories: scattering reactions and absorption [6]. We define the *scattering cross-section* as follows:

$$\sigma_\mathrm{s} = \sigma_\mathrm{e} + \sigma_\mathrm{in} + \sum_{x \geqslant 2} x\, \sigma_{\mathrm{n,xn}} \tag{2.8}$$

where σ_e is the *elastic (or potential) scattering cross-section*, σ_in is the *inelastic cross-section* and $\sigma_{\mathrm{n},x\mathrm{n}\ x\geqslant 2}$ the reactions that emit more than 2 neutrons but without undergoing a fission.

The *absorption cross-section* σ_a is defined as the sum of all other types of reactions: fission reactions (σ_f), radiative capture (σ_γ), and transmutation reactions (σ_α: an α particle is emitted; σ_p: a proton is emitted, etc.).

Except the potential scattering, all nuclear reactions involve the formation of a compound nucleus and their cross-sections may exhibit high variations with neutron energy [7, 6]. Therefore, all other quantities depending on cross-sections like the mean free path will also exhibit high variations with the neutron energy.

Figure 2.1 shows for example the evolution of the total cross-section of Pu^{239} with the incident neutron energy. This figure was created using the on-line cross-section plotter of `http://atom.kaeri.re.kr` and the Jeff2.2 library.

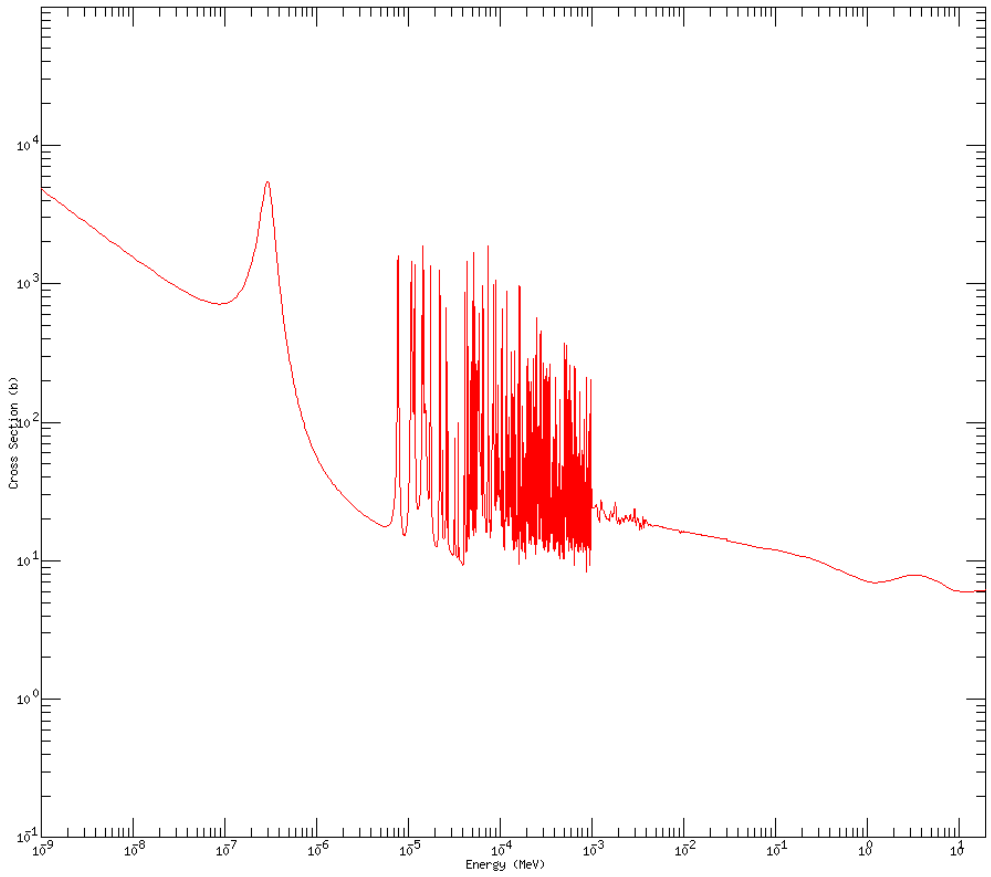


Figure 2.1 – Total cross-section of ^{239}Pu in barn, function of the incident neutron energy (MeV)

We observe three regions: the thermal region at low energies where the cross-section is regular, the epithermal region which exhibits several resonances with important variations, and the high energies where resonances are no more resolved: here we are only able to give an average value of the cross-sections, but it does not mean that there are no more resonances.

2.2 The particle flux

In the reactor, each particle is described in Cartesian coordinates by:

- three position coordinates $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$
- three velocity coordinates $\vec{v} = v_x\vec{i} + v_y\vec{j} + v_z\vec{k}$ and $\vec{v} = \frac{\mathrm{d}\vec{r}}{\mathrm{d}t}$. Practically, we use the velocity module $V_n = \|\vec{v_n}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ and the solid angle $\vec{\Omega}$ which is defined by $\vec{\Omega} = \frac{1}{\|\vec{v_n}\|} \cdot \vec{v} = \mu\vec{i} + \eta\vec{j} + \epsilon\vec{k}$ and represents the unit vector of direction of movement. This normalization leads to the fact that only two components of $\vec{\Omega}$ are independent: for example the *colatitude* or *polar angle* ψ and the *azimuth* ϕ which determines all components using:

$$\begin{cases} \mu^2 + \eta^2 + \epsilon^2 = 1 \\ \mu = \cos\psi \\ \eta = \cos\phi \\ \epsilon = \sin\phi \end{cases} \Leftrightarrow \begin{cases} \eta^2 + \epsilon^2 = 1 - \mu^2 = \sin^2\psi \\ \mu = \cos\psi \\ \eta = \sqrt{1-\mu^2}\cos\phi \\ \epsilon = \sqrt{1-\mu^2}\sin\phi \end{cases} \tag{2.9}$$

Let us consider an elemental volume d^3r. A population of particles is represented by a distribution called the population density $n(\vec{r}, V_n, \vec{\Omega}, t)$ such that $n(\vec{r}, V_n, \vec{\Omega}, t)\,\mathrm{d}^3r\,\mathrm{d}V_n\,\mathrm{d}^2\Omega$ is the number of particles at time t, in the volume element d^3r surrounding position $\vec{r}$, in the velocity element $\mathrm{d}V_n$ surrounding V_n and in the solid angle element $\mathrm{d}^2\Omega$ surrounding $\vec{\Omega}$.

We define the *angular flux* as a distribution related to the population density and its velocity:

$$\phi(\vec{r}, V_n, \vec{\Omega}, t) = n(\vec{r}, V_n, \vec{\Omega}, t) \cdot V_n \tag{2.10}$$

We may want to sum the flux on all angular directions. By performing a distribution reduction on variable $\vec{\Omega}$, we define the *direction integrated flux* $\phi(\vec{r}, V_n, t)$:

$$\phi(\vec{r}, V_n, t) = \int_{4\pi} \phi(\vec{r}, V_n, \vec{\Omega}, t)\,\mathrm{d}^2\Omega \tag{2.11}$$

In order to spare notations, we use the same symbol to represent the angular flux and the direction integrated flux. The presence or absence of $\vec{\Omega}$ in the arguments will tell us which quantity we are considering. The same convention was adopted in [6].

We commonly use the following change of variable for V_n:

$$\begin{aligned} E &= \frac{1}{2} m V_n^2 \quad \text{with} \quad \mathrm{d}E = m V_n \, \mathrm{d}V_n \\ u &= \ln(\frac{E_0}{E}) \quad \text{with} \quad \mathrm{d}u = \left| \frac{E}{E_0} E_0 \left(-\frac{1}{E^2} \right) \right| \mathrm{d}E = \frac{1}{E} \, \mathrm{d}E \end{aligned} \tag{2.12}$$

where E is the energy of the particle, m its mass and V_n the norm of its velocity vector; u is a quantity called the lethargy, with E_0 the maximum energy of a particle so that u remains positive. It gives:

$$\begin{aligned} \phi(\vec{r}, E, t) &= \frac{1}{m V_n} \phi(\vec{r}, V_n, t), \;\; E_0 \geqslant E > 0 \\ \phi(\vec{r}, u, t) &= E \phi(\vec{r}, E, t), \;\; u \geqslant 0 \, . \end{aligned} \tag{2.13}$$

The angular flux and the integrated flux do not have a precise physical meaning [6]. In fact, the physical quantity that makes sense is the angular current of particle $\vec{J}$. It represents the number $\frac{\mathrm{d}N}{\mathrm{d}t}$ of particles of velocity V_n passing through an elemental surface $\mathrm{d}^2 S$ per unit of time [7].

$$\begin{aligned} \frac{\mathrm{d}N}{\mathrm{d}t} &= \underbrace{n(\vec{r}, V_n, \vec{\Omega}, t) \, V_n \, \vec{\Omega}}_{\vec{J}(\vec{r}, V_n, \vec{\Omega}, t)} \cdot \vec{N} \, \mathrm{d}^2 S \\ \Rightarrow \quad \vec{J}(\vec{r}, V_n, \vec{\Omega}, t) &= \phi(\vec{r}, V_n, \vec{\Omega}, t) \, \vec{\Omega} \end{aligned} \tag{2.14}$$

We can define the direction integrated current:

$$\vec{J}(\vec{r}, V_n, t) = \int_{4\pi} \vec{J}(\vec{r}, V_n, \vec{\Omega}, t) \, \mathrm{d}^2\Omega = \int_{4\pi} \phi(\vec{r}, V_n, \vec{\Omega}, t) \, \vec{\Omega} \, \mathrm{d}^2\Omega \tag{2.15}$$

2.3 The transport equation

The transport equation is the mathematical translation of the principle of conservation of the particles.

2.3.1 The differential form of the transport equation

Let us define a *control volume* $\mathcal{V}$ surrounded by a *controlled surface* $\partial\mathcal{V}$. We consider particles located in $\mathcal{V}$, travelling in direction $\vec{\Omega}$ within a $\mathrm{d}^2\Omega$ interval, with a velocity equal

to V_n within a $\mathrm{d}V_n$ interval. The initial number of particle is equal to:

$$\int_{\mathcal{V}} n(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}^3 r. \tag{2.16}$$

This number is going to change during an infinitesimal time $\mathrm{d}t$ according to following balance relation:

$$\begin{aligned}&(\text{Number of particles at } t + \mathrm{d}t \text{ in } \mathcal{V}) - (\text{Number of particles at } t \text{ in } \mathcal{V})\\ &= (\text{Number of particle created in } \mathcal{V}) - (\text{Number of particles lost by collision in } \mathcal{V})\\ &\quad + (\text{Particles entering } \mathcal{V} \text{ through } \partial\mathcal{V}) - (\text{Particles coming out of } \mathcal{V} \text{ through } \partial\mathcal{V}).\end{aligned} \tag{2.17}$$

Mathematically, the terms of the previous equation are written:

– The variation of the particle number:

$$\mathrm{d}^3\mathcal{A} = \int_{\mathcal{V}} \Big[n(\vec{r}, V_n, \vec{\Omega}, t + \mathrm{d}t) - n(\vec{r}, V_n, \vec{\Omega}, t) \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}^3 r \tag{2.18}$$

– The net number of particles streaming out of $\mathcal{V}$ during $\mathrm{d}t$:

$$\mathrm{d}^3\mathcal{B} = \int_{\partial\mathcal{V}} \phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \cdot \vec{N}\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^2 r \tag{2.19}$$

where $\vec{N}$ is the unit vector normal to the surface $\partial\mathcal{V}$, pointing outside $\partial\mathcal{V}$. The surface $\partial\mathcal{V}$ is located at position $\vec{r}$. A negative value would mean that in reality, more particles are coming in than streaming out of $\mathcal{V}$. With the divergence theorem [10], this expression is equal to:

$$\mathrm{d}^3\mathcal{B} = \int_{\mathcal{V}} \mathrm{div} \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r \tag{2.20}$$

$$= \int_{\mathcal{V}} \vec{\nabla} \cdot \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r \tag{2.21}$$

– The number of collisions in $\mathrm{d}^3 r$ during $\mathrm{d}t$ is equal to:

$$\begin{aligned}\mathrm{d}^3\mathcal{C} &= \int_{\mathcal{V}} \Sigma(\vec{r}, V_n) \Big[n(\vec{r}, V_n, \vec{\Omega}, t) \cdot V_n \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r\\ &= \int_{\mathcal{V}} \Sigma(\vec{r}, V_n) \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t) \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r\end{aligned} \tag{2.22}$$

where we assume that the total macroscopic cross-section Σ is independent of $\vec{\Omega}$ and t. This hypothesis simplifies the notations, but it is usually not valid. When a collision

occurs, we consider that the particle will always gain or loose some speed, or take another direction. Therefore it vanishes from our integration element because it streams out of $\mathrm{d}V_n$.

– The number of new particles created in $\mathcal{V}$ during $\mathrm{d}t$ is equal to:

$$\mathrm{d}^3\mathcal{D} = \int_{\mathcal{V}} Q(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r \tag{2.23}$$

where Q is the distribution representing the source of neutrons, which we will define more precisely later.

The transport equation is then written:

$$\mathrm{d}^3\mathcal{A} = -\mathrm{d}^3\mathcal{B} - \mathrm{d}^3\mathcal{C} + \mathrm{d}^3\mathcal{D} \tag{2.24}$$

Replacing the terms by their expressions, we obtain:

$$\begin{aligned}
&\int_{\mathcal{V}} \Big[n(\vec{r}, V_n, \vec{\Omega}, t + \mathrm{d}t) - n(\vec{r}, V_n, \vec{\Omega}, t) \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}^3 r \\
&= -\int_{\mathcal{V}} \vec{\nabla} \cdot \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r \\
&\quad - \int_{\mathcal{V}} \Sigma(\vec{r}, V_n)\, \phi(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r \\
&\quad + \int_{\mathcal{V}} Q(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t\, \mathrm{d}^3 r
\end{aligned} \tag{2.25}$$

The transport equation is valid for all control volumes $\mathcal{V}$, so that we can discard it in the four terms and obtain the differential form of this equation:

$$\begin{aligned}
&\Big[n(\vec{r}, V_n, \vec{\Omega}, t + \mathrm{d}t) - n(\vec{r}, V_n, \vec{\Omega}, t) \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega \\
&= -\vec{\nabla} \cdot \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \Big]\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t \\
&\quad - \Sigma(\vec{r}, V_n)\, \phi(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t \\
&\quad + Q(\vec{r}, V_n, \vec{\Omega}, t)\, \mathrm{d}V_n\, \mathrm{d}^2\Omega\, \mathrm{d}t
\end{aligned} \tag{2.26}$$

Finally, we divide by $\mathrm{d}t$, $\mathrm{d}V_n$, $\mathrm{d}^2\Omega$ and take the limit as $\mathrm{d}t \to 0$ so that we recognize a partial derivation of $n(\vec{r}, V_n, \vec{\Omega}, t)$. We obtain:

$$\frac{\partial n(\vec{r}, V_n, \vec{\Omega}, t)}{\partial t} = -\vec{\nabla} \cdot \Big[\phi(\vec{r}, V_n, \vec{\Omega}, t)\, \vec{\Omega} \Big] - \Sigma(\vec{r}, V_n)\, \phi(\vec{r}, V_n, \vec{\Omega}, t) + Q(\vec{r}, V_n, \vec{\Omega}, t) \tag{2.27}$$

We then use a formula to expand the divergence term:

$$\mathrm{div}\left(\phi\,\vec{\Omega}\right) = \underbrace{\phi\,\mathrm{div}\left(\vec{\Omega}\right)}_{=0} + \vec{\Omega}\cdot\overrightarrow{\mathrm{grad}}\left(\phi\right) \tag{2.28}$$

This leads to the final formulation, introducing the flux in the first term:

$$\frac{1}{V_n}\frac{\partial\phi(\vec{r},V_n,\vec{\Omega},t)}{\partial t} = -\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},V_n,\vec{\Omega},t)\right] - \Sigma(\vec{r},V_n)\,\phi(\vec{r},V_n,\vec{\Omega},t) + Q(\vec{r},V_n,\vec{\Omega},t) \tag{2.29}$$

Notice that the term V_n does not depend on t, which explains why it can be moved out of the partial derivation.

In steady-state conditions, all partial time derivatives upon t vanish. The equation becomes:

$$\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},V_n,\vec{\Omega})\right] + \Sigma(\vec{r},V_n)\,\phi(\vec{r},V_n,\vec{\Omega}) = Q(\vec{r},V_n,\vec{\Omega}) \tag{2.30}$$

This form is suitable for certain methods of resolution such as the P_N or S_N methods. The streaming operator $\vec{\Omega}\cdot\vec{\nabla}$ has to be explicitly written depending on the coordinate system and then discretized. However, we can also transform this equation in two other equivalent forms, leading to two other methods of solution.

2.3.2 The characteristic form of the transport equation

We define a *characteristic* as a straight line of direction $\vec{\Omega}$ corresponding to a neutron trajectory. We then define the curvilinear abscissa s as the distance s of the neutron on a characteristic from a reference position $\vec{r}$ on this characteristic. Therefore, we can consider a change of variables:

$$\begin{cases} \vec{r}\,(t) = \vec{r}\,(t_0) + s\vec{\Omega} \\ t = t_0 + \dfrac{s}{V_n} \end{cases} \Rightarrow \begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}s} = \vec{\Omega}\cdot\vec{\nabla} + \dfrac{1}{V_n}\dfrac{\partial}{\partial t} \end{cases} \tag{2.31}$$

The previous implication is relatively easy to demonstrate with Cartesian coordinates:

$$\begin{aligned}
&\begin{cases} \mathrm{d}s\,\vec{\Omega} = \mathrm{d}\vec{r} = \mathrm{d}x\vec{i} + \mathrm{d}y\vec{j} + \mathrm{d}z\vec{k} \\ \dfrac{\mathrm{d}}{\mathrm{d}s} = \dfrac{\mathrm{d}x}{\mathrm{d}s}\dfrac{\partial}{\partial x} + \dfrac{\mathrm{d}y}{\mathrm{d}s}\dfrac{\partial}{\partial y} + \dfrac{\mathrm{d}z}{\mathrm{d}s}\dfrac{\partial}{\partial z} + \dfrac{1}{V_n}\dfrac{\partial}{\partial t} \end{cases} \\
\Rightarrow \quad & \frac{\mathrm{d}}{\mathrm{d}s} = \left(\vec{\Omega}\cdot\vec{i}\right)\frac{\partial}{\partial x} + \left(\vec{\Omega}\cdot\vec{j}\right)\frac{\partial}{\partial y} + \left(\vec{\Omega}\cdot\vec{k}\right)\frac{\partial}{\partial z} + \frac{1}{V_n}\frac{\partial}{\partial t} \\
\Leftrightarrow \quad & \frac{\mathrm{d}}{\mathrm{d}s} = \vec{\Omega}\cdot\vec{\nabla} + \frac{1}{V_n}\frac{\partial}{\partial t}
\end{aligned} \tag{2.32}$$

It is more complicated with other coordinate systems, because we need to explicitly calculate the Jacobian matrix.

Substituting equation 2.31 in 2.29 gives the *backward characteristic form of the transport equation*:

$$\frac{\mathrm{d}\phi(\vec{r}+s\vec{\Omega},V_n,\vec{\Omega},t+s/V_n)}{\mathrm{d}s} + \Sigma(\vec{r}+s\vec{\Omega},V_n)\,\phi(\vec{r},V_n,\vec{\Omega},t+s/V_n) = Q(\vec{r}+s\vec{\Omega},V_n,\vec{\Omega},t+s/V_n). \tag{2.33}$$

It can also be written in the forward form as:

$$-\frac{\mathrm{d}\phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)}{\mathrm{d}s} + \Sigma(\vec{r}-s\vec{\Omega},V_n)\,\phi(\vec{r},V_n,\vec{\Omega},t-s/V_n) = Q(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n). \tag{2.34}$$

This equation will be solved by the method of characteristics [11]. But we can continue to manipulate this equation to provide an integral form.

2.3.3 The integral form of the transport equation

Let us define the *optical path* as:

$$\tau(s,V_n) = \int_0^s \Sigma(\vec{r}-s'\vec{\Omega},V_n)\,\mathrm{d}s' \tag{2.35}$$

We start by computing the following expression:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}s}\left[e^{-\tau(s,V_n)} \times \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] \\
&= e^{-\tau(s,V_n)}\frac{\mathrm{d}}{\mathrm{d}s}\left[\phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] \\
&\qquad + \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\frac{\mathrm{d}}{\mathrm{d}s}\left[e^{-\tau(s,V_n)}\right] \\
&= e^{-\tau(s,V_n)}\frac{\mathrm{d}}{\mathrm{d}s}\left[\phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] \\
&\qquad + \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n) \times e^{-\tau(s,V_n)}\left(-\Sigma(\vec{r}-s\vec{\Omega})\right) \\
&= e^{-\tau(s,V_n)}\Bigg(\frac{\mathrm{d}}{\mathrm{d}s}\left[\phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] \\
&\qquad + \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\left(-\Sigma(\vec{r}-s\vec{\Omega})\right)\Bigg)
\end{aligned} \tag{2.36}$$

where we used the identity:

$$\frac{\mathrm{d}}{\mathrm{d}s}\int_0^s f(s')\,\mathrm{d}s' = f(s). \tag{2.37}$$

We recognize part of the characteristic forward form of the transport equation. Substitution of 2.34 into 2.36 gives:

$$-\frac{\mathrm{d}}{\mathrm{d}s}\left[e^{-\tau(s,V_n)} \times \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] = e^{-\tau(s,V_n)}\left[Q(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right] \tag{2.38}$$

We then perform an integration between 0 and $+\infty$ over ds:

$$\begin{aligned}
&-\int_0^{+\infty}\frac{\mathrm{d}}{\mathrm{d}s}\left[e^{-\tau(s,V_n)} \times \phi(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right]\mathrm{d}s \\
&\qquad = \int_0^{+\infty} e^{-\tau(s,V_n)}\left[Q(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right]\mathrm{d}s
\end{aligned} \tag{2.39}$$

$$\Leftrightarrow \quad \phi(\vec{r},V_n,\vec{\Omega},t) = \int_0^{+\infty} e^{-\tau(s,V_n)}\left[Q(\vec{r}-s\vec{\Omega},V_n,\vec{\Omega},t-s/V_n)\right]\mathrm{d}s \tag{2.40}$$

Equation 2.40 is the integral form of the transport equation for an infinite domain [6]. Physically, it represents the neutron flux generated by a source at distance s from a position $\vec{r}$. The exponential term represents the attenuation of the sources with the distance s: the further is the source, the fewer neutrons can reach the position $\vec{r}$. The attenuation coefficient is the total cross-section of each mixture laying between the source and the studied position.

This form of the transport equation is the basis for the collision probability method of

resolution [6].

In the case of a finite domain, the infinite integral must be replaced by a finite one to take into account the fact that the characteristics are finite. We obtain:

$$\begin{aligned}\phi(\vec{r}, V_n, \vec{\Omega}, t) = e^{-\tau(b,V_n)}\, \phi(\vec{r} - b\vec{\Omega}, V_n, \vec{\Omega}, t - b/V_n) \\ + \int_0^b e^{-\tau(s,V_n)} \left[Q(\vec{r} - s\vec{\Omega}, V_n, \vec{\Omega}, t - s/V_n) \right] \mathrm{d}s\end{aligned} \tag{2.41}$$

where b is the maximal distance that a neutron can cover on one characteristic [6]. An integration has then to be performed for all characteristics to get the flux in one position. The term $e^{-\tau(b,V_n)}\, \phi(\vec{r} - b\vec{\Omega}, V_n, \vec{\Omega}, t - b/V_n)$ can be seen as the boundary flux, and is therefore related to the boundary conditions.

2.4 The boundary conditions

Several boundary conditions can be defined depending on our models. Let us call $\partial\mathcal{V}$ the boundary surface of volume $\mathcal{V}$, $\vec{r}_s$ a position on this surface, $\vec{N}(\vec{r}_s)$ the outward normal to this surface at $\vec{r}_s$. Setting a boundary condition means to clarify the behaviour of the incoming flux $\phi(\vec{r}_s, V_n, \vec{\Omega}, t)$ for all incoming directions: $\vec{\Omega} \cdot \vec{N}(\vec{r}_s) < 0$ [6]. We can use:

– the albedo boundary condition:

$$\phi(\vec{r}_s, V_n, \vec{\Omega}, t) = \beta \phi(\vec{r}_s, V_n, \vec{\Omega}', t) \text{ with } \vec{\Omega} \cdot \vec{N}(\vec{r}_s) < 0 \tag{2.42}$$

where $\vec{\Omega}'$ is the direction of the outgoing particle. $\beta = 1$ corresponds to a reflective condition whereas $\beta = 0$ corresponds to a vacuum condition.

– The specular reflection corresponds to the case where:

$$\vec{\Omega} \cdot \vec{N}(\vec{r}_s) = -\vec{\Omega}' \cdot \vec{N}(\vec{r}_s) \text{ and } \left(\vec{\Omega} \wedge \vec{\Omega}\right) \cdot \vec{N}(\vec{r}_s) = 0. \tag{2.43}$$

This necessitate a tracking of the geometry where tracks are undergoing a mirror reflection when they reach edges. The implementation is quite difficult and requires a lot of computational resources.

– The white boundary condition is easier to implement and is therefore often used because it is a good approximation of the mirror reflection. The white reflective condition considers that all particles reaching the edge turn back to the volume with an isotropic

angular distribution: for $\vec{\Omega} \cdot \vec{N}(\vec{r}_s) < 0$ we have:

$$\phi(\vec{r}_s, V_n, \vec{\Omega}, t) = \beta \cdot \frac{\displaystyle\int_{\vec{\Omega}\cdot\vec{N}(\vec{r}_s)>0} \left[\vec{\Omega}' \cdot \vec{N}(\vec{r}_s)\right] \phi(\vec{r}_s, V_n, \vec{\Omega}', t)\, \mathrm{d}^2\Omega'}{\displaystyle\int_{\vec{\Omega}\cdot\vec{N}(\vec{r}_s)>0} \left[\vec{\Omega}' \cdot \vec{N}(\vec{r}_s)\right] \mathrm{d}^2\Omega'}$$

$$\Leftrightarrow \quad \phi(\vec{r}_s, V_n, \vec{\Omega}, t) = \frac{\beta}{\pi} \cdot \int_{\vec{\Omega}\cdot\vec{N}(\vec{r}_s)>0} \left[\vec{\Omega}' \cdot \vec{N}(\vec{r}_s)\right] \phi(\vec{r}_s, V_n, \vec{\Omega}', t)\, \mathrm{d}^2\Omega'. \tag{2.44}$$

– The periodic boundary condition expresses the fact that the lattice is infinite and periodic: the flux on one boundary is equal to the flux on another parallel boundary:

$$\phi(\vec{r}_s, V_n, \vec{\Omega}, t) = \phi(\vec{r}_s + \delta\vec{r}, V_n, \vec{\Omega}, t). \tag{2.45}$$

where $\delta\vec{r}$ is the lattice pitch.

Finally, we can add that inside the domain $\mathcal{V}$, the angular flux $\phi(\vec{r}, V_n, \vec{\Omega}, t)$ must be continuous across all internal interfaces in the direction $\vec{\Omega}$ of the moving neutron. On the contrary, continuity is not required along directions which are not parallel to the path of travel.

2.5 The steady-state source density

In a reactor, the sources of neutrons with an energy E are mainly:
– the neutrons created by the fissions;
– the neutrons coming from the scattering reactions;
– the neutrons created by (n, xn) reactions.

This section will focus on the term $Q(\vec{r}, E, \vec{\Omega})$ of the transport equation. We use the variable of energy E instead of V_n in our equations. As our study is based on steady-state conditions, we will not consider the time dependence of the equations. More information about transient analysis can be found in [12, 6].

2.5.1 The scattering source

The scattering reactions can be considered as a source of neutrons: indeed, an incident neutron of energy E' which is undergoing such a reaction can produce a secondary neutron with an energy E. We call this source $Q^{\mathrm{scat}}(\vec{r}, E, \vec{\Omega})$ and we write:

$$Q^{\mathrm{scat}}(\vec{r}, E, \vec{\Omega}) = \int_{4\pi} \int_0^{+\infty} \Sigma_{\mathrm{s}}(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}')\, \phi(\vec{r}, E', \vec{\Omega}')\, \mathrm{d}E'\, \mathrm{d}^2\Omega' \tag{2.46}$$

where $\Sigma_s(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}')$ is the macroscopic differential scattering cross-section taking into account diffusion and (n, xn) reactions. It means that for each type of reaction, we have to know the probability of a neutron having a collision at a certain energy E' with a collision direction $\vec{\Omega}'$, and exiting at energy E in the direction $\vec{\Omega}$.

Most media can be considered as isotropic, because the neutron mass is by far inferior to the mass of the collided nuclei. A well-known exception is water [7], which contains hydrogen nuclei whose mass is close to that of a neutron. But this assumption is quite reasonable for graphite reflectors [8] and for the fuel mixtures. In this case, the scattering cross-section is only a function of the scattering angle:

$$\Sigma_s(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}') = \frac{1}{2\pi}\Sigma_s(\vec{r}, E \leftarrow E', \vec{\Omega} \cdot \vec{\Omega}') \tag{2.47}$$

It is then convenient to write an expansion with Legendre polynomials:

$$\Sigma_s(\vec{r}, E \leftarrow E', \vec{\Omega} \cdot \vec{\Omega}') = \sum_{l=0}^{L} \frac{2l+1}{2}\Sigma_{s,l}(\vec{r}, E \leftarrow E')P_l(\vec{\Omega} \cdot \vec{\Omega}') \tag{2.48}$$

where L is the scattering order of the medium considered. $L = 0$ and $L = 1$ correspond to isotropic scattering and to linearly anisotropic scattering in the LAB, respectively [6]. The Legendre coefficients $\Sigma_{s,l}(E \leftarrow E')$ are defined as:

$$\Sigma_{s,l}(E \leftarrow E') = \int_{-1}^{1} \Sigma_s(E \leftarrow E', \mu)P_l(\mu)\,\mathrm{d}\mu \tag{2.49}$$

We can also expand the flux with spherical harmonics to approximate the integral over $\mathrm{d}^2\Omega$, so that we obtain [6]:

$$Q^{\mathrm{scat}}(\vec{r}, E, \vec{\Omega}) = \int_0^{+\infty} \sum_{l=0}^{L} \frac{2l+1}{4\pi}\Sigma_{s,l}(\vec{r}, E \leftarrow E') \sum_{m=-l}^{l} R_l^m(\vec{\Omega})\,\phi_l^m(\vec{r}, E')\,\mathrm{d}E' \tag{2.50}$$

where

$$\phi_l^m(\vec{r}, E') = \int_{4\pi} R_l^m(\vec{\Omega})\phi(\vec{r}, E, \vec{\Omega})\,\mathrm{d}^2\Omega. \tag{2.51}$$

2.5.2 The fission source

The fission source is due to the fission reactions which produce some secondary neutrons. It is usually isotropic in the LAB because they involve a compound nucleus which disintegrates much later after its formation. Therefore, it does not conserve any memory of the incoming direction of the neutron [6].

Thus, we write:

$$Q^{\text{fiss}}(\vec{r}, E, \vec{\Omega}) = \frac{1}{4\pi K_{\text{eff}}} Q^{\text{fiss}}(\vec{r}, E) \tag{2.52}$$

K_{eff} is called the effective multiplication factor. It enables us to do a steady-state calculation even if the reactor is not at the equilibrium, which would mean that the sum of absorption and leakage equals the production rate of new fission neutrons [6]. The K_{eff} adjusts the fission source so that we go back to steady-state conditions. If $K_{\text{eff}} > 1$ the reactor is supercritical: the production rate of neutrons is higher than the loss rate of neutrons, and the population of neutrons is increasing. On the contrary, when $K_{\text{eff}} < 1$, losses are more important than production of neutrons, and the population is decreasing. The reactor is said to be subcritical. With $K_{\text{eff}} = 1$, equilibrium is reached: the reactor is critical. The determination of this constant is therefore very important in the reactor analysis for safety, operation and design [7].

We assume that the isotropic fission source is independent of the energy of the incident neutron. However, the emitted neutrons do not have always the same energy. This information is given by the fission spectrum $\chi_i(E)$, different for each fissile nuclide *i*. $\chi_i(E)$ is the probability for an emitted neutron to have an energy equal to E within a $\mathrm{d}E$ interval in the LAB [6]. As it is a probability, it is normalised to 1:

$$\int_0^{+\infty} \chi_i(E)\, \mathrm{d}E = 1. \tag{2.53}$$

Thus, the isotropic fission source is written:

$$Q^{\text{fiss}}(\vec{r}, E) = \sum_{j=1}^{J^{\text{fiss}}} \chi_j(E) \int_0^{+\infty} \nu \Sigma_{\text{f},j}(\vec{r}, E') \phi(\vec{r}, E')\, \mathrm{d}E' \tag{2.54}$$

where:

J^{fiss}: total number of fissile isotopes

ν: number of emitted neutrons per fission, taken as constant here

$\Sigma_{\text{f},j}(\vec{r}, E)$: macroscopic fission cross-section of the j$^{\text{th}}$ fissile isotope.

2.5.3 Complete source expression

When we add the two source terms, we obtain [6]:

$$Q(\vec{r}, E, \vec{\Omega}) = \int_0^{+\infty} \sum_{l=0}^{L} \frac{2l+1}{4\pi} \Sigma_{\mathrm{s},l}(\vec{r}, E \leftarrow E') \sum_{m=-l}^{l} R_l^m(\vec{\Omega})\, \phi_l^m(\vec{r}, E')\, \mathrm{d}E' + \frac{1}{4\pi K_{\mathrm{eff}}} \sum_{j=1}^{J^{\mathrm{fiss}}} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r}, E')\phi(\vec{r}, E')\, \mathrm{d}E' \tag{2.55}$$

We can remark that:

$$\phi(\vec{r}, E) = \phi_0^0(\vec{r}, E). \tag{2.56}$$

The substitution of this term in the transport equation 2.30 leads to the singular form of the transport equation. As stated in [6], it is an Eigenvalue problem:

- The neutron flux appears in each term. $\phi = 0$ is a trivial solution. A set of non-trivial solutions (Eigenvectors or Eigensolutions) exists for some discrete values of K_{eff}. The possible K_{eff} values are the Eigenvalues and are associated with eigenvectors ϕ. The fundamental solution corresponds to the maximum possible value of K_{eff} and is the only Eigensolution with a physical meaning [6].
- The flux distribution corresponding to the fundamental solution is positive or null everywhere. All other solutions are harmonics and become negative in some regions. A linear combination of the fundamental Eigensolution and of the different harmonics can lead to a solution which is positive or null everywhere, which would have a physical meaning. However, we always use the approximation that the flux is well represented by the fundamental solution, without corrections by the harmonics [6].
- The flux distribution of each eigensolution can be arbitrarily normalized. The value of the normalization constant is usually computed from the thermal power P of the reactor that is measured:

  $$P = \int_0^{+\infty} \int_{\mathcal{V}} H(\vec{r}, E)\phi(\vec{r}, E)\, \mathrm{d}^3 r\, \mathrm{d}E. \tag{2.57}$$

  where $\mathcal{V}$ is the volume of the reactor and $H(\vec{r}, E)$ is the power factor giving the recoverable energy in terms of the flux.

We recall here that in theory, from a neutronics point of view, any power can be achieved by any reactor if it is able to become supercritical. The neutronic limitation only occurs, for example, if the increase of the temperature changes the cross-sections and provides a negative feed-back strong enough to decrease K_{eff} under 1. But the power does not depend directly on the resolution of the steady-state transport equation.

2.6 The transport correction

Solving the transport equation with linear anisotropy in the scattering source is difficult. The method of collision probabilities has large difficulties dealing with anisotropic sources. Therefore, it has been imagined to "trick" the cross-sections, so that a calculation in isotropic condition would take into account some linear anisotropic effects. The principle is to add a forward-peaked component in the Legendre expansion of the differential scattering cross-sections [6]. This special treatment of the cross-sections is not strictly mathematically correct, but it is however widely used. This additional component takes the form of a Dirac delta term:

$$\Sigma_{\mathrm{s}}(\vec{r}, E \leftarrow E', \mu) = \sum_{l=0}^{L} \frac{2l+1}{2} \bar{\Sigma}_{s,l}(\vec{r}, E \leftarrow E') P_l(\mu) + \Delta\Sigma_{\mathrm{tr}}(\vec{r}, E')\delta(E - E')\delta(\mu - 1) \quad (2.58)$$

where $\bar{\Sigma}_{s,l}(\vec{r}, E \leftarrow E')$ is a modified Legendre coefficient and $\Delta\Sigma_{\mathrm{tr}}(\vec{r}, E')$ is the additional coefficient multiplying the Dirac delta term. They are computed so as to preserve the Legendre moments:

$$\forall\, l \in [\![0; L+1]\!], \quad \Sigma_{\mathrm{s},l}(\vec{r}, E \leftarrow E') = \int_{-1}^{1} \Sigma_{\mathrm{s}}(\vec{r}, E \leftarrow E', \mu) P_l(\mu)\, \mathrm{d}\mu. \quad (2.59)$$

Using $P_l(1) = 1$, we obtain:

$$\begin{cases} \forall\, l \in [\![0; L]\!], \quad \bar{\Sigma}_{s,l}(\vec{r}, E \leftarrow E') + \Delta\Sigma_{\mathrm{tr}}(\vec{r}, E')\delta(E - E') = \Sigma_{\mathrm{s},l}(\vec{r}, E \leftarrow E') \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Delta\Sigma_{\mathrm{tr}}(\vec{r}, E') = \Sigma_{\mathrm{s},L+1}(\vec{r}, E') \end{cases} \quad (2.60)$$

Then, writing $L = 0$, we have:

$$\Sigma_{\mathrm{s}}(\vec{r}, E \leftarrow E', \mu) = \frac{1}{2}\left[\Sigma_{\mathrm{s},0}(\vec{r}, E \leftarrow E') - \Sigma_{\mathrm{s},1}(\vec{r}, E')\delta(E - E')\right] + \Sigma_{\mathrm{s},1}(\vec{r}, E')\delta(E - E')\delta(\mu - 1) \quad (2.61)$$

We then substitute in the transport equation 2.30:

$$\vec{\Omega} \cdot \vec{\nabla}\phi(\vec{r}, E, \vec{\Omega}) + \bar{\Sigma}(\vec{r}, E)\phi(\vec{r}, E, \vec{\Omega}) = \bar{Q}(\vec{r}, E, \vec{\Omega}) \quad (2.62)$$

where we define the transport-corrected macroscopic total cross-section $\bar{\Sigma}(\vec{r}, E)$ by:

$$\bar{\Sigma}(\vec{r}, E) = \Sigma(\vec{r}, E) - \Sigma_{\mathrm{s},1}(\vec{r}, E) \quad (2.63)$$

and where the transport-corrected steady-state source is now written:

$$\begin{aligned}\bar{Q}(\vec{r},E,\vec{\Omega}) =& \frac{1}{4\pi}\int_0^{+\infty}\left[\Sigma_{\mathrm{s},0}(\vec{r},E\leftarrow E') - \Sigma_{\mathrm{s},1}(\vec{r},E')\delta(E-E')\right]\phi(\vec{r},E')\,\mathrm{d}E' \\ &+ \frac{1}{4\pi K_{\mathrm{eff}}}\sum_{j=1}^{J^{\mathrm{fiss}}}\chi_j(E)\int_0^{+\infty}\nu\Sigma_{\mathrm{f},j}(\vec{r},E')\phi(\vec{r},E')\,\mathrm{d}E' \end{aligned} \tag{2.64}$$

With the transport correction, the source density remains mathematically isotropic, while the transport equation includes a correction for the anisotropic scattering [6]. It is therefore particularly interesting for lattice codes.

2.7 The multi-group steady-state transport equation

The variation of the cross-sections as a function of the energy of the incoming neutron is analytically not known. Therefore, it is necessary to discretize the energy domain in intervals in which we will consider that the cross-sections are constant and independent of energy [7]. Those intervals are called groups. They can be of various width. The more groups we take, the better we will be able to approach the real behaviour of the cross-sections. However, a compromise has to be struck between the accuracy of this representation and the computational limitations (time, memory). Optimizing the number of groups for a calculation and the position of the boundaries is a huge task in itself, outside the scope of this work, and several propositions can be made. Usually, deterministic codes use between 150 and 400 groups for the lattice calculations, and less than 50 groups for the whole core calculations [6]. An energy condensation is performed between the two calculations to reduce the number of groups. We also assume a separation of the energy dependence from the space and angular dependence within each group and within the region of interest.

We write below the transport equation in its three forms with G groups:

$$\begin{aligned} & \forall g \in [\![1;G]\!], \\ & \vec{\Omega}\cdot\vec{\nabla}\left[\phi_g(\vec{r},\vec{\Omega})\right] + \Sigma_g(\vec{r})\phi_g(\vec{r},\vec{\Omega}) = Q_g(\vec{r},\vec{\Omega}) && (2.65) \\ \Leftrightarrow\quad & \frac{\mathrm{d}\phi_g(\vec{r}+s\vec{\Omega},\vec{\Omega})}{\mathrm{d}s} + \Sigma_g(\vec{r}+s\vec{\Omega})\,\phi_g(\vec{r},\vec{\Omega}) = Q_g(\vec{r}+s\vec{\Omega},\vec{\Omega}) && (2.66) \\ \Leftrightarrow\quad & \phi_g(\vec{r},\vec{\Omega}) = \int_0^{+\infty} e^{-\tau_g(s)}\left[Q_g(\vec{r}-s\vec{\Omega},\vec{\Omega})\right]\mathrm{d}s && (2.67) \end{aligned}$$

with the optical path in group g:

$$\tau_g(s) = \int_0^s \Sigma_g(\vec{r}-s'\vec{\Omega})\,\mathrm{d}s' \tag{2.68}$$

and the source density which couples the groups between themselves through the terms $\Sigma_{\mathrm{s},l,g\leftarrow h}(\vec{r})$ and $\chi_{j,g}$:

$$Q_g(\vec{r},\vec{\Omega}) = \sum_{h=1}^{G}\sum_{l=0}^{L}\frac{2l+1}{4\pi}\Sigma_{\mathrm{s},l,g\leftarrow h}(\vec{r})\sum_{m=-l}^{l}R_l^m(\vec{\Omega})\phi_{l,h}^m(\vec{r}) + \frac{1}{4\pi K_{\mathrm{eff}}}\sum_{j=1}^{J^{\mathrm{fiss}}}\chi_{j,g}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},j,h}(\vec{r})\phi_h(\vec{r}) \tag{2.69}$$

The multigroup transport correction leads to:

$$\forall g \in [\![1;G]\!], \quad \Delta\Sigma_{\mathrm{tr},g}(\vec{r}) = \Sigma_{\mathrm{s},1,g}(\vec{r}) \tag{2.70}$$

The multigroup transport corrected macroscopic cross-section is defined by:

$$\forall g \in [\![1;G]\!], \quad \bar{\Sigma}_g(\vec{r}) = \Sigma_g(\vec{r}) - \Delta\Sigma_{\mathrm{tr},g}(\vec{r}) \tag{2.71}$$

and the multigroup P_0 transport-corrected component of the differential scattering cross-section is written:

$$\forall (g,h) \in [\![1;G]\!]^2, \quad \bar{\Sigma}_{\mathrm{s},0,g\leftarrow h}(\vec{r}) = \Sigma_{\mathrm{s},0,g\leftarrow h}(\vec{r}) - \delta_{gh}\,\Delta\Sigma_{\mathrm{tr},g}(\vec{r}) \tag{2.72}$$

where δ_{gh} is the Kronecker delta function. Equation 2.72 is likely to produce negative $\bar{\Sigma}_{\mathrm{s},0,g\leftarrow g}(\vec{r})$ components.[6]

2.8 The collision probability method

The collision probability method is the first method of resolution of the transport equation that was historically used in reactor physics and it uses the integral from of the transport equation (see equation 2.40). This method is still used because of its relative robustness and speed. A detailed presentation of this method can be found in [6]. It was chosen in this thesis to focus rather on the method of characteristics.

2.9 The method of characteristics

This method uses the characteristic form of the transport equation:

$$\begin{aligned}&\forall g \in [\![1; G]\!],\\&\frac{\mathrm{d}\phi_g(\vec{r}+s\vec{\Omega},\vec{\Omega})}{\mathrm{d}s} + \Sigma_g(\vec{r}+s\vec{\Omega})\,\phi_g(\vec{r},\vec{\Omega}) = Q_g(\vec{r}+s\vec{\Omega},\vec{\Omega})\end{aligned} \tag{2.73}$$

The method of characteristics (MOC) consists of tracking several characteristics over the domain and following a neutron travelling through an entire characteristic from one side to the other side [11]. On the contrary, the CP method considers a point and counts the number of neutrons able to reach it from the other regions. But the two methods have a similar philosophy and, for example, we can use the same tracking file [6].

The MOC algorithm proceeds in three steps [6]:

1. The geometry is tracked, which means that we draw a sufficient number of characteristics over the entire geometry. Typical tracking parameters are the number of angular directions for the tracks and the density of parallel tracks on a perpendicular segment per centimetre. Each time the characteristic encounters a boundary or a change of composition, it delimits a segment. The tracking obtained can be used with the CP method or with the MOC. In the case of a finite domain, the tracks are cut at the boundaries where we set boundary conditions.
2. The integration of the algebraic collapsing (AC) matrices is performed by assuming that the cross- sections and the flux are constant on each segment. Parallelization is possible to compute each matrix in each group, because there are no interactions between them. AC matrices are required for the synthetic acceleration of the MOC. [11]
3. Finally, the angular flux is calculated by iterations and is integrated over solid angles.

Again, like in the P_N method, the outgoing angular fluxes are expanded on an orthogonal basis of flux functions $\{\psi_i(\vec{\Omega}\cdot\vec{N})\}_{i\in\mathbb{N}}$ with $\vec{\Omega}\cdot\vec{N} > 0$ where $\vec{N}$ is the outgoing normal of the considered cell.

We usually limit ourselves to the orders 0 or 1. Cutting the expansion at the order 0 means that we consider an isotropic outgoing current. The first order introduces the possibility to have some anisotropy in the direction of the current, which is much more accurate and usually preferable.

The characteristics are crossing different regions, and therefore are cut into K pieces, K being an integer representing the total number of regions crossed by one characteristic [11].

We consider the average flux on each segment and the average source:

$$\begin{aligned}\phi_{k,g} &= \frac{1}{l_k}\int_0^{l_k} \phi_g(\vec{r_k} + s\vec{\Omega})\,\mathrm{d}s \\ Q_{k,g} &= \frac{1}{l_k}\int_0^{l_k} Q_g(\vec{r_k} + s\vec{\Omega}, \vec{\Omega})\,\mathrm{d}s\end{aligned} \tag{2.74}$$

where $\vec{r_k}$ is the point of entry of the characteristic inside the region k. Thus, we have $\vec{r_{k+1}} = \vec{r_k} + l_k\vec{\Omega}$.

We consider a constant total cross-section $\Sigma_{k,g}$ and also usually a constant isotropic source on the segment:

$$\forall s \in [0, l_k], \quad Q_g(\vec{r_k} + s\vec{\Omega}, \vec{\Omega}) = \frac{Q_{k,g}}{4\pi} \tag{2.75}$$

With this approximation known as the step-characteristic approximation [11], we can integrate analytically the transport equation over one segment:

$$\begin{aligned}&\frac{\mathrm{d}\phi_{k,g}(\vec{r} + s\vec{\Omega}, \vec{\Omega})}{\mathrm{d}s} + \Sigma_{k,g}(\vec{r} + s\vec{\Omega})\,\phi_{k,g}(\vec{r}, \vec{\Omega}) = \frac{Q_{k,g}}{4\pi}\,l_k \\ \Rightarrow \quad &\exists\ \phi_0 \in \mathbb{R}\ /\ \phi_{k,g}(s) = \phi_0\, e^{-\Sigma_{k,g}s} + \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} l_k\end{aligned} \tag{2.76}$$

With the help of the boundary condition, we find the constant ϕ_0:

$$\begin{aligned}&\phi_{k,g}(s=0) = \phi_{k,g} = \phi_0 + \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} l_k \\ \Leftrightarrow \quad &\phi_0 = \phi_{k,g} - \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} l_k \\ \Rightarrow \quad &\phi_{k,g}(s) = \left(\phi_{k,g} - \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} l_k\right) e^{\Sigma_{k,g}s} + \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} l_k\end{aligned} \tag{2.77}$$

It leads to $\phi_{k+1,g}$ for $s = l_k$:

$$\phi_{k+1,g} = \phi_{k,g} e^{-\Sigma l_k} + \frac{Q_k}{4\pi\Sigma_{k,g}} l_k \left(1 - e^{-\Sigma_{k,g} l_k}\right) \tag{2.78}$$

We can also discretize the derivation operator of the transport equation assuming that l_k is small and replacing the flux by a constant inside the segment l_k. It leads to another relation [11]:

$$\phi_{k+1,g} - \phi_{k,g} + \Sigma_{k,g} l_k \phi_{k,g} = \frac{Q_{k,g}}{4\pi} \tag{2.79}$$

where $\phi_{k,g}$ is the average of the flux on the segment.

When we replace $\phi_{k+1,g}$ by its expression in function of $\phi_{k,g}$, we obtain:

$$\phi_{k,g} = \frac{\phi_{k,g}}{\Sigma_{k,g} l_k} \left(1 - e^{-\Sigma_{k,g} l_k}\right) + \frac{Q_{k,g}}{4\pi\Sigma_{k,g}} \left(1 - \frac{1 - e^{-\Sigma_{k,g} l_k}}{\Sigma_{k,g} l_k}\right) \tag{2.80}$$

The source term still depends on the flux. Equations 2.79 and 2.80 will have to be solved by iterations, taking care of the fact that the optical path $\tau_{k,g} = \Sigma_{k,g} l_k$ must remain small.

The interest is that it produces matrices of size $N_L \times L$ where N_L is the number of regions and L the number of interfaces. The collision probability method generates matrices of size $N_L \times N_L$, and for large domains we have $N_L > L$ [11]. Therefore, the MOC is often recommended to do calculations over large domains if memory must be spared. But on the contrary, the CP algorithm reads the tracking once and then computes the matrices of collision probabilities, whereas the MOC algorithm needs to read the tracking at each iteration, which decreases the calculation speed. It explains the importance of preconditioning and acceleration methods for MOC in order to converge within a reasonable time [11].

2.10 The discrete ordinates method

The S_n method starts with the differential form of the transport equation by discretizing the angular variable $\vec{\Omega}$ into n directions [8]. The letter S stands for straight lines [6]. Each direction is characterized by a direction cosine. Those are chosen to maximize the accuracy of the integration: a quadrature is used to weight the importance of each direction, like for example a classical Gauss-Legendre quadrature [6]. The method provides a convenient way to discretize the streaming operator $\vec{\Omega} \cdot \vec{\nabla}\phi(\vec{r}, \vec{\Omega})$. This is very efficient, but it was not available for hexagonal geometries in DRAGON [3]. However, we have been using it on a cylindrical geometry for our reflector calculations.

2.11 Other elements of a lattice code

2.11.1 Neutron slowing down and resonance self-shielding

The neutrons produced by fission reactions are generally emitted at a high energy. At those energies, their probability to lead a nuclide to undergo a fission is low. They usually undergo scattering reactions until they reach the thermal energy domain, where the fission cross-sections of the fissile isotopes are much higher. Up-scattering can only happen in the thermal domain. It describes the fact that a neutron which has been slowed down below the thermal agitation energy can regain some energy while encountering a nuclide. This phenomenon is limited to the thermal energies: it cannot take a neutron back into the

highest energies [6]. The neutron slowing down is essentially the consequence of collisions with light nuclides like water or graphite, which are put intentionally in the reactor to increase the fission probabilities. Such materials are called moderators [7]. The slowing down effect of heavy nuclides is small because the mass difference between a single neutron and heavy nuclides prevents them from being efficient moderators. We can write a simplified version of the transport equation without the fission source called the slowing-down equation, where we separate the two slowing-down terms according to the type of nuclides:

$$\vec{\Omega} \cdot \vec{\nabla}\phi(\vec{r}, u, \vec{\Omega}) + \Sigma(\vec{r}, u)\phi(\vec{r}, u, \vec{\Omega}) = \frac{1}{4\pi}\left[\mathcal{R}^{+}\left(\phi(\vec{r}, u, \vec{\Omega})\right) + \mathcal{R}^{*}\left(\phi(\vec{r}, u, \vec{\Omega})\right)\right] \tag{2.81}$$

where:

$$\begin{cases} \mathcal{R}^{+}\left(\phi(\vec{r}, u, \vec{\Omega})\right) = \text{slowing-down operator for light nuclides} \\ \mathcal{R}^{*}\left(\phi(\vec{r}, u, \vec{\Omega})\right) = \text{slowing-down operator for heavy nuclides} \end{cases}$$

$$\Rightarrow \quad \begin{cases} \mathcal{R}^{+}\left(\phi(\vec{r}, u, \vec{\Omega})\right) = \displaystyle\int_{0}^{+\infty} \Sigma_{s0}^{+}(\vec{r}, u \leftarrow u', \vec{\Omega})\phi(\vec{r}, u')\,\mathrm{d}u' \\ \mathcal{R}^{*}\left(\phi(\vec{r}, u, \vec{\Omega})\right) = \displaystyle\int_{0}^{+\infty} \Sigma_{s0}^{*}(\vec{r}, u \leftarrow u', \vec{\Omega})\phi(\vec{r}, u')\,\mathrm{d}u' \end{cases}$$

During the slowing down process, the absorption cross-sections of the heavy nuclides stay low on average, but they show narrow, very large peaks at certain energies in the epithermal domain (see figure 2.1 page 11). Those peaks are called resonances. A neutron coming out of a scattering reaction with an energy located right on a resonance will have a very high probability of being absorbed. Therefore, the flux will be depressed within every resonance, due to the fact that most neutrons arriving at those energies through the scattering reactions will be absorbed [6, 7]. However and fortunately, the resonances are usually much narrower than the lethargy gain of each scattering reaction so that overall, only a small number of neutrons are absorbed and the flux depressions remain small [7]. Paul Reuss describes this phenomenon in his book with a wonderful analogy involving kangaroos: the lethargy interval gained by a neutron at each scattering reaction during the slowing down is much bigger than the lethargy width of a resonance, so that a neutron has generally a large probability of skipping the resonance region in three or four jumps, as a kangaroo would happily jump above deep but narrow traps [7].

A first solution to deal with the resonances would be to discretize the energy variable enough, so that we arrive at the experimental curve of the cross-sections. Some codes have adopted this way, in particular the codes preparing the cross-section libraries like NJOY99. But it is not feasible to apply this solution in neutronics codes because it would require

more than 12 000 groups of energy: any calculation would have prohibitive costs of time and memory, and the convergence would be very difficult to achieve.

With a coarser energy mesh, for example with 295 groups [13], it is not possible to describe all resonances in the epithermal domain. Taking an average without precaution would result in a stronger average flux than the one observed because resonances are not described. This phenomenon is called resonance self-shielding [7]. The retained solution is to adjust the cross-sections to recover somehow the average flux that would have been calculated with a very detailed energy mesh describing the resonances. The cross-sections which lead to the correct average flux in the resonance regions are called self-shielded cross-sections. They are calculated by the formula:

$$\bar{\sigma}_{\rho,g} = \mu_g \frac{\int_{u_{g-1}}^{u_g} \sigma_{\rho,g}\phi(u)\,\mathrm{d}u}{\int_{u_{g-1}}^{u_g} \phi(u)\,\mathrm{d}u} \tag{2.82}$$

where μ_g is a super homogenisation factor which stands to correct the geometrical differences, and u the lethargy.

This procedure conserves the reaction rate that would be obtained with an exact calculation of the flux. The difficulty is that of course we do not know this exact flux. To solve this problem, Livolant and Jeanpierre have proposed a series of approximations [14, 6]. The first one is that the effect of the resonances would be the same whatever the geometry: therefore, the flux can be separated into two pieces:

$$\phi(\vec{r}, u, \vec{\Omega}) = \varphi(\vec{r}, u, \vec{\Omega}) \times \psi(\vec{r}, u) \tag{2.83}$$

where $\varphi(\vec{r}, u, \vec{\Omega})$ is a resonant fine-structure function carrying the resonance informations, and $\psi(\vec{r}, u)$ the macroscopic flux that would be observed without resonances. It is also the asymptotic behaviour of the flux between the resonances.

The macroscopic flux is affected by the slowing down operator for nuclear reactions with non-resonant isotopes, but not by the slowing-down operator for resonant nuclides. On the contrary, the fine-structure function is only affected by the resonant operator. We write:

$$\begin{aligned} \mathcal{R}^+ \left(\phi(\vec{r}, u, \vec{\Omega})\right) &= \Sigma^+_{\mathrm{s0}}(\vec{r}, u)\psi(\vec{r}, u) \\ \mathcal{R}^* \left(\phi(\vec{r}, u, \vec{\Omega})\right) &= \psi(\vec{r}, u)\mathcal{R}^* \left(\varphi(\vec{r}, u)\right) \end{aligned} \tag{2.84}$$

Another approximation consists of assuming that the macroscopic flux is spatially flat:

$$\vec{\Omega} \cdot \vec{\nabla}\phi(\vec{r}, u, \vec{\Omega}) = \psi(\vec{r}, u, \vec{\Omega}) \left[\vec{\Omega} \cdot \vec{\nabla}\varphi(\vec{r}, u, \vec{\Omega})\right] \tag{2.85}$$

This approximation enables us to simplify the macroscopic flux from the slowing-down

equation and obtain an equation for the fine-structure function which will be solved using more approximations. The fine-structure function will then be used to compute the self-shielded cross-sections:

$$\begin{cases} \vec{\Omega}\cdot\vec{\nabla}\varphi(\vec{r},u,\vec{\Omega}) + \Sigma(\vec{r},u)\varphi(\vec{r},u) = \dfrac{1}{4\pi}\left[\Sigma_{\mathrm{s}}^{+}(\vec{r},u) + \mathcal{R}^{*}\left(\varphi(\vec{r},u)\right)\right] \\ \bar{\sigma}_{\rho,g} = \mu_g \cdot \dfrac{\displaystyle\int_{u_{g-1}}^{u_g} \sigma_{\rho,g}\varphi(u)\,\mathrm{d}u}{\displaystyle\int_{u_{g-1}}^{u_g} \varphi(u)\,\mathrm{d}u} \end{cases} \tag{2.86}$$

Solving equation 2.86 requires approximations on the slowing-down term before using classical tools of solution of the transport equation. Several possibilities have been tried, and it is still an object of interesting research. One of them is called probability tables and has been widely used in our DRAGON calculations [6, 3]. It consists of replacing a Riemann integral by a Lebesgue integral and then approximating it by Dirac Delta functions. Mathematically, we have:

$$\begin{cases} \underbrace{\dfrac{1}{u_g - u_{g-1}}\displaystyle\int_{u_{g-1}}^{u_g} \sigma_{\rho,g}\,\varphi(\sigma_{\rho,g}(u))\,\mathrm{d}u}_{\text{Riemann integral}} = \underbrace{\displaystyle\int_{0}^{\max(\sigma_{\rho,g})} \Pi(\sigma)\sigma_{\rho,g}\,\varphi(\sigma)\,\mathrm{d}\sigma}_{\text{Lebesgue integral}} \\ \text{Approximation: } \Pi(\sigma)\,\sigma_{\rho,g}\,\varphi(\sigma) \approx \displaystyle\sum_{k=1}^{K} \delta(\sigma-\sigma_{g,k})\,\omega_k\,\sigma_{\rho,g,k}\,\phi(\sigma_{g,k}) \end{cases}$$

$$\Rightarrow \quad \frac{1}{u_g - u_{g-1}}\int_{u_{g-1}}^{u_g} \sigma_{\rho,g}\,\varphi(\sigma_{\rho,g}(u))\,\mathrm{d}u \approx \sum_{k=1}^{K} \delta(\sigma-\sigma_{g,k})\,\omega_k\,\sigma_{\rho,g,k}\,\phi(\sigma_{g,k}) \tag{2.87}$$

The set of values $\{\omega_k, \sigma_{k,g}\}_{k\in[\![1;K]\!]}$ is the probability table of order K for group g, and for the reaction and nuclide considered. The ω_k constitutes a quadrature set. More information about their establishment and the details of this method can be found in [6, 15, 16]. This method is currently considered to be one of the best to calculate the self-shielded cross-sections.

The self-shielding step is very sensitive in a calculation scheme, because it directly affects the cross-sections used for the solution of the transport equation. It can potentially create significant discrepancies between different codes. It is also one of the major differences between deterministic codes and Monte-Carlo codes which use continuous energy group structures.

2.11.2 The homogenisation and condensation: SPH equivalence technique

Once the transport equation has been solved, lattice codes are asked to produce homogenized cross-sections for full-core calculations. In addition, full-core calculations are usually done with fewer groups to gain speed. Therefore, a condensation has to be performed, for example from 295 groups to 26 groups.

Merging the cross-sections over different regions requires one to establish a macro-balance relation between the detailed cross-sections and the merged one. We perform an average of those cross-sections weighted by the flux. This procedure is called the flux-volume homogenization [17].

Let us define a collection of N regions of volumes V_i, $i \in [\![1; N]\!]$. The volume of the merged region is defined by:

$$V_{\mathrm{m}} = \sum_{i=1}^{N} V_i \tag{2.88}$$

Let us consider that the lattice calculation has been done with a set of G_g groups, $g \in [\![1; G]\!]$ and that we would like to condense it to M groups, $M \ll G$. We define M_k as a collection of several groups $g \in [\![1; G]\!]$, so that $\bigcup_{k=1}^{M} M_k = \bigcup_{g=1}^{G} G_g$.

A lattice calculation provides the heterogeneous fluxes ($\phi_{i,g}$) in each group $g \in [\![1; G]\!]$ and region $i \in [\![1; N]\!]$. The homogenized cross-sections of the merged region V_{m} are calculated by the following formula for reaction x:

$$\bar{\Sigma}_{\mathrm{x},m,k} = \frac{\left.\sum_{g \in M_k} \sum_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g} \right\} \text{Reaction rate}}{\left.\sum_{g \in M_k} \sum_{i=1}^{N} V_i\, \phi_{i,g} \right\} \text{Integrated flux}} \tag{2.89}$$

Theoretically, this procedure conserves the reaction rates calculated by the lattice calculation. This is achieved only if, when using the merged cross-sections, the full-core calculation produces for each coarse energy group k a homogeneous flux $\Phi_{\mathrm{hom},k}$ equal to the integrated flux calculated above [17]. But in fact, this situation is not very likely to occur. In this case, the full-core calculation is producing a homogeneous flux $\Phi_{\mathrm{hom},k}$ so that we have:

$$\begin{aligned} \Phi_{\mathrm{hom},k} &\neq \sum_{g \in M_k} \sum_{i=1}^{N} V_i\, \phi_{i,g} \\ \bar{\Sigma}_{\mathrm{x},m,k}\, \Phi_{\mathrm{hom},k} &\neq \sum_{g \in M_k} \sum_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g} \end{aligned} \tag{2.90}$$

If we think that the reaction rates provided by the lattice code are better than those obtained in the full-core calculation using the simple flux-volume homogenization, then it is possible to force the conservation of those reaction rates by doing a super homogenisation (SPH) [6, 17].

It consists in adding a constant μ_k called SPH-factor during the homogenization:

$$\tilde{\Sigma}_{\mathrm{x},m,k} = \underbrace{\mu_k}_{\text{SPH-factor}} \frac{\sum\limits_{g\in M_k}\sum\limits_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g}}{\sum\limits_{g\in M_k}\sum\limits_{i=1}^{N} V_i\, \phi_{i,g}} \tag{2.91}$$

This SPH-factor is determined so that we preserve the reaction rates:

$$\begin{aligned}
& \tilde{\Sigma}_{\mathrm{x},m,k}\, \Phi_{\mathrm{hom},k} = \sum_{g\in M_k}\sum_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g} \\
\Leftrightarrow \quad & \mu_k \frac{\sum\limits_{g\in M_k}\sum\limits_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g}}{\sum\limits_{g\in M_k}\sum\limits_{i=1}^{N} V_i\, \phi_{i,g}} \cdot \Phi_{\mathrm{hom},k} = \sum_{g\in M_k}\sum_{i=1}^{N} \Sigma_{\mathrm{x},i,g}\, V_i\, \phi_{i,g} \\
\Leftrightarrow \quad & \mu_k = \frac{\sum_{g\in M_k}\sum_{i=1}^{N} V_i\, \phi_{i,g}}{\Phi_{\mathrm{hom},k}}
\end{aligned} \tag{2.92}$$

The algorithm starts at iteration $i = 1$ with: $\forall\, k \in M_k$, $\mu_k^1 = 1$. It produces a first set of homogeneous fluxes in each coarse group $\left(\Phi^1_{\mathrm{hom},k}\right)_{k\in M_k}$. Then we compute the second set of SPH-factors, use them to correct the homogenized macroscopic cross-sections and redo the full-core calculation. This second iteration produces a second set of homogeneous fluxes and then a third set of SPH-factors. We iterate on the process until the SPH factors are converged:

$$\forall \epsilon \in \mathbb{R}^+,\ \exists\ i \in \mathbb{N}^*\ /\ \forall k \in [\![1; M]\!] \quad \left| \frac{\mu_k^{i+1} - \mu_k^i}{\mu_k^i} \right| \leqslant \epsilon \tag{2.93}$$

This procedure is of course time-consuming, because we have to do the full-core calculation several times before reaching convergence. However, the full-core calculation may be fast and when it converges, the SPH homogenisation ensures that we correctly preserve the reaction rates of the lattice calculation at the core level so that the solution may be much more accurate. This is particularly true regarding the power shape in the core.

2.12 Full-core calculation: the P_N method

Two classes of methods are used to perform whole core calculations within a relatively short calculation time: the diffusion codes and the transport P_N codes. Other transport methods would require too much time and memory to be efficient.

Diffusion codes necessitate the calculation of a diffusion coefficient after the lattice calculation. This is done using leakage models and the fundamental mode approximation. This strategy is well suited for reactors moderated by water because each assembly is quite decoupled from their neighbours. The mean free paths of the thermal neutrons are small regarding the dimensions of the assemblies, so that the fundamental mode approximation is well verified and the diffusion codes give good results. But for reactors with higher mean free paths like fast reactors or graphite moderated reactors, the diffusion approximations may be less suitable. It explains why transport codes were developed for core analysis purposes to avoid the diffusion approximation.

The P_N method is based on an expansion of the streaming operator with spherical harmonics and Legendre polynomials. The integer N of P_N represents the order of truncation of the series.

2.12.1 The P_1 equations

We will first explain the P_1 equations with only one group. The derivation in multi-group theory is similar and will be developed after. To understand the derivation, one must keep in mind the following identities [18]:

$$\int_{4\pi} \mathrm{d}^2\Omega = 4\pi \tag{2.94}$$

$$\int_{4\pi} \vec{\Omega}\, \mathrm{d}^2\Omega = \vec{0} \tag{2.95}$$

$$\int_{4\pi} \vec{A}\cdot\vec{\Omega}\, \mathrm{d}^2\Omega = 0 \tag{2.96}$$

$$\int_{4\pi} \left(\vec{\Omega}\cdot\vec{A}\right)\vec{\Omega}\, \mathrm{d}^2\Omega = \frac{4\pi}{3}\vec{A} \tag{2.97}$$

$$\int_{4\pi} \left(\vec{\Omega}\cdot\vec{A}\right)\left(\vec{\Omega}\cdot\vec{B}\right)\, \mathrm{d}^2\Omega = \frac{4\pi}{3}\vec{A}\cdot\vec{B} \tag{2.98}$$

$$\int_{4\pi} \left(\vec{\Omega}\cdot\vec{A}\right)\left(\vec{\Omega}\cdot\vec{B}\right)\vec{\Omega}\, \mathrm{d}^2\Omega = \vec{0}. \tag{2.99}$$

where $\vec{A}$ and $\vec{B}$ are two vectors that do not depend on $\vec{\Omega}$.

We approximate the angular flux by:

$$\phi(\vec{r},E,\vec{\Omega}) = \frac{1}{4\pi}\phi_0(\vec{r},E) + \frac{3}{4\pi}\vec{\Omega}\cdot\vec{\phi_1}(\vec{r},E). \tag{2.100}$$

We remark that those coefficients have a clear physical meaning [8]:

$$\begin{aligned}
\phi(\vec{r},E) &= \int_{4\pi}\phi(\vec{r},E,\vec{\Omega})\,\mathrm{d}^2\Omega && (2.101)\\
&= \frac{1}{4\pi}\phi_0(\vec{r},E)\underbrace{\int_{4\pi}\mathrm{d}^2\Omega}_{4\pi} + \frac{3}{4\pi}\underbrace{\int_{4\pi}\vec{\phi_1}(\vec{r},E)\cdot\vec{\Omega}\,\mathrm{d}^2\Omega}_{0} && (2.102)\\
&= \phi_0(\vec{r},E) && (2.103)\\
\vec{J}(\vec{r},E) &= \int_{4\pi}\phi(\vec{r},E,\vec{\Omega})\,\vec{\Omega}\,\mathrm{d}^2\Omega && (2.104)\\
&= \frac{1}{4\pi}\phi_0(\vec{r},E)\underbrace{\int_{4\pi}\vec{\Omega}\,\mathrm{d}^2\Omega}_{0} + \frac{3}{4\pi}\underbrace{\int_{4\pi}\left(\vec{\phi_1}(\vec{r},E)\cdot\vec{\Omega}\right)\vec{\Omega}\,\mathrm{d}^2\Omega}_{\frac{4\pi}{3}\vec{\phi_1}(\vec{r},E)} && (2.105)\\
&= \vec{\phi_1}(\vec{r},E) && (2.106)
\end{aligned}$$

Therefore, we will redefine directly the angular flux with the physical expression of the coefficients:

$$\phi(\vec{r},E,\vec{\Omega}) = \frac{1}{4\pi}\phi(\vec{r},E) + \frac{3}{4\pi}\vec{\Omega}\cdot\vec{J}(\vec{r},E). \tag{2.107}$$

We transfer this approximation in the transport equation and we integrate over 4π with infinitesimal volume $\mathrm{d}^2\Omega$:

$$\begin{aligned}
&\underbrace{\int_{4\pi}\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right]\,\mathrm{d}^2\Omega}_{I_1} + \underbrace{\int_{4\pi}\Sigma(\vec{r},E)\,\phi(\vec{r},E,\vec{\Omega})\,\mathrm{d}^2\Omega}_{I_2}\\
&\qquad = \underbrace{\int_{4\pi}\int_{4\pi}\int_0^{+\infty}\Sigma_\mathrm{s}(\vec{r},E\leftarrow E',\vec{\Omega}\leftarrow\vec{\Omega}')\,\phi(\vec{r},E',\vec{\Omega}')\,\mathrm{d}E'\,\mathrm{d}^2\Omega'\,\mathrm{d}^2\Omega}_{I_3}\\
&\qquad\quad + \underbrace{\frac{1}{4\pi K_\mathrm{eff}}\sum_{j=1}^{J^\mathrm{fiss}}\int_{4\pi}\chi_j(E)\int_0^{+\infty}\nu\Sigma_{\mathrm{f},j}(\vec{r},E')\phi(\vec{r},E',\vec{\Omega})\,\mathrm{d}E'\,\mathrm{d}^2\Omega}_{I_4}.
\end{aligned} \tag{2.108}$$

We will then deal with each term separately.

$$\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right] = \vec{\Omega}\cdot\vec{\nabla}\left[\frac{1}{4\pi}\phi(\vec{r},E)+\frac{3}{4\pi}\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right] \tag{2.109}$$

$$= \frac{1}{4\pi}\vec{\Omega}\cdot\vec{\nabla}\phi(\vec{r},E)+\frac{3}{4\pi}\vec{\Omega}\cdot\vec{\nabla}\left[\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right] \tag{2.110}$$

$$I_1 = \int_{4\pi}\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right]\,\mathrm{d}^2\Omega \tag{2.111}$$

$$= \frac{1}{4\pi}\vec{\nabla}\phi(\vec{r},E)\cdot\underbrace{\int_{4\pi}\vec{\Omega}\,\mathrm{d}^2\Omega}_{=\vec{0}}+\frac{3}{4\pi}\int_{4\pi}\vec{\Omega}\cdot\vec{\nabla}\left[\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right]\,\mathrm{d}^2\Omega \tag{2.112}$$

$$= \frac{3}{4\pi}\int_{4\pi}\vec{\Omega}\cdot\vec{\nabla}\left[\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right]\,\mathrm{d}^2\Omega \tag{2.113}$$

$$\Leftrightarrow\ I_1 = \frac{3}{4\pi}\int_0^1\int_{4\pi}\mathrm{div}\left[\left(\vec{J}(\vec{r},E)\cdot\vec{\Omega}\right)\vec{\Omega}\right]\,\mathrm{d}^2\Omega\,\rho^2\,\mathrm{d}\rho$$

using the Green-Ostrogradski theorem

$$= \frac{3}{4\pi}\int_{4\pi}\left(\vec{J}(\vec{r},E)\cdot\vec{\Omega}\right)\vec{\Omega}\cdot\vec{\Omega}\,\mathrm{d}^2\Omega$$

using the Green-Ostrogradski theorem in the other direction

$$= \frac{3}{4\pi}\int_{4\pi}\left(\vec{J}(\vec{r},E)\cdot\vec{\Omega}\right)\underbrace{\left(\vec{\Omega}\cdot\vec{\Omega}\right)}_{=1}\,\mathrm{d}^2\Omega$$

$$= \frac{3}{4\pi}\int_0^1\int_{4\pi}\mathrm{div}\vec{J}(\vec{r},E)\,\mathrm{d}^2\Omega\,\rho^2\,\mathrm{d}\rho$$

$$= \frac{3}{4\pi}\mathrm{div}\vec{J}(\vec{r},E)\underbrace{\int_0^1\int_{4\pi}\mathrm{d}^2\Omega\,\rho^2\,\mathrm{d}\rho}_{=\frac{4\pi}{3}}$$

$$= \mathrm{div}\vec{J}(\vec{r},E). \tag{2.114}$$

$$
\begin{aligned}
I_2 &= \int_{4\pi} \Sigma(\vec{r}, E)\, \phi(\vec{r}, E, \vec{\Omega})\, \mathrm{d}^2\Omega \\
&= \Sigma \int_{4\pi} \frac{1}{4\pi}\phi(\vec{r}, E) + \frac{3}{4\pi}\vec{\Omega} \cdot \vec{J}(\vec{r}, E)\, \mathrm{d}^2\Omega \\
&= \frac{1}{4\pi}\Sigma\phi(\vec{r}, E) \int_{4\pi} \mathrm{d}^2\Omega \\
&= \Sigma\phi(\vec{r}, E).
\end{aligned} \tag{2.115}
$$

For the scattering source, we also assume a P_1 approximation:

$$
\Sigma_\mathrm{s}(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}') = \frac{1}{4\pi}\Sigma_{\mathrm{s}0}(\vec{r}, E \leftarrow E') + \frac{3}{4\pi}\Sigma_{\mathrm{s},1}(\vec{r}, E \leftarrow E') \quad \vec{\Omega} \cdot \vec{\Omega}'. \tag{2.116}
$$

Using this, we have:

$$
\begin{aligned}
I_3 &= \int_{4\pi}\int_{4\pi}\int_0^{+\infty} \Sigma_\mathrm{s}(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}')\, \phi(\vec{r}, E', \vec{\Omega}')\, \mathrm{d}E'\, \mathrm{d}^2\Omega'\, \mathrm{d}^2\Omega \\
&= \int_{4\pi}\int_{4\pi}\int_0^{+\infty} \left[\frac{1}{4\pi}\Sigma_{\mathrm{s}0}(\vec{r}, E \leftarrow E') + \frac{3}{4\pi}\Sigma_{\mathrm{s},1}(\vec{r}, E \leftarrow E') \quad \vec{\Omega} \cdot \vec{\Omega}'\right] \times \\
&\qquad\qquad \left[\frac{1}{4\pi}\phi(\vec{r}, E') + \frac{3}{4\pi}\vec{\Omega}' \cdot \vec{J}(\vec{r}, E')\right] \mathrm{d}E'\, \mathrm{d}^2\Omega'\, \mathrm{d}^2\Omega \\
&= \frac{1}{4\pi}\int_0^{+\infty}\int_{4\pi} \Sigma_{\mathrm{s}0}(E \leftarrow E')\phi(\vec{r}, E')\, \mathrm{d}E'\, \mathrm{d}^2\Omega \\
&= \int_0^{+\infty} \Sigma_{\mathrm{s}0}(E \leftarrow E')\phi(\vec{r}, E')\, \mathrm{d}E'
\end{aligned} \tag{2.117}
$$

Indeed, we remark that:

$$
\int_{4\pi}\int_{4\pi} \vec{\Omega} \cdot \vec{\Omega}'\, \mathrm{d}^2\Omega\, \mathrm{d}^2\Omega' = \underbrace{\int_{4\pi} \vec{\Omega}\, \mathrm{d}^2\Omega}_{=\vec{0}} \cdot \underbrace{\int_{4\pi} \vec{\Omega}'\, \mathrm{d}^2\Omega'}_{=\vec{0}} = \vec{0}. \tag{2.118}
$$

For I_4, we just integrate the angular flux, which leads to:

$$
\begin{aligned}
I_4 &= \frac{1}{4\pi K_\mathrm{eff}} \sum_{j=1}^{J^\mathrm{fiss}} \int_{4\pi} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r}, E')\phi(\vec{r}, E')\, \mathrm{d}E'\, \mathrm{d}^2\Omega \\
&= \frac{1}{K_\mathrm{eff}} \sum_{j=1}^{J^\mathrm{fiss}} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r}, E')\phi(\vec{r}, E')\, \mathrm{d}E'
\end{aligned} \tag{2.119}
$$

Finally, we obtain a first equation:

$$\begin{aligned}\mathrm{div}\vec{J}(\vec{r},E)+\Sigma\phi(\vec{r},E) &= \int_0^{+\infty}\Sigma_{\mathrm{s}0}(E\leftarrow E')\phi(\vec{r},E')\,\mathrm{d}E' \\ &\quad+\frac{1}{K_{\mathrm{eff}}}\sum_{j=1}^{J^{\mathrm{fiss}}}\chi_j(E)\int_0^{+\infty}\nu\Sigma_{\mathrm{f},j}(\vec{r},E')\phi(\vec{r},E')\,\mathrm{d}E'.\end{aligned} \tag{2.120}$$

A second, vectorial equation can be obtained by multiplying the transport equation by $\vec{\Omega}$ before integrating over 4π on $\mathrm{d}^2\Omega$:

$$\begin{aligned}&\underbrace{\int_{4\pi}\left(\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right]\right)\vec{\Omega}\,\mathrm{d}^2\Omega}_{I_1}+\underbrace{\int_{4\pi}\Sigma(\vec{r},E)\,\phi(\vec{r},E,\vec{\Omega})\,\vec{\Omega}\,\mathrm{d}^2\Omega}_{I_2}\\ &=\underbrace{\int_{4\pi}\int_{4\pi}\int_0^{+\infty}\Sigma_{\mathrm{s}}(\vec{r},E\leftarrow E',\vec{\Omega}\leftarrow\vec{\Omega}')\,\phi(\vec{r},E',\vec{\Omega}')\,\vec{\Omega}\,\mathrm{d}E'\,\mathrm{d}^2\Omega'\,\mathrm{d}^2\Omega}_{I_3}\\ &\quad+\underbrace{\frac{1}{4\pi K_{\mathrm{eff}}}\sum_{j=1}^{J^{\mathrm{fiss}}}\int_{4\pi}\chi_j(E)\int_0^{+\infty}\nu\Sigma_{\mathrm{f},j}(\vec{r},E')\int_{4\pi}\phi(\vec{r},E',\vec{\Omega}')\,\vec{\Omega}\,\mathrm{d}E'\,\mathrm{d}^2\Omega'\,\mathrm{d}^2\Omega}_{I_4}.\end{aligned} \tag{2.121}$$

In this case, the simplification of the streaming operator part is not totally obvious.

$$\begin{aligned}I_1&=\int_{4\pi}\left(\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right]\right)\vec{\Omega}\,\mathrm{d}^2\Omega\\ &=\int_{4\pi}\left(\vec{\Omega}\cdot\vec{\nabla}\left[\frac{1}{4\pi}\phi(\vec{r},E)+\frac{3}{4\pi}\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right]\right)\vec{\Omega}\,\mathrm{d}^2\Omega\\ &=\underbrace{\frac{1}{4\pi}\int_{4\pi}\left[\vec{\Omega}\cdot\vec{\nabla}\left(\phi(\vec{r},E)\right)\right]\vec{\Omega}\,\mathrm{d}^2\Omega}_{I_a}+\underbrace{\frac{3}{4\pi}\int_{4\pi}\left[\vec{\Omega}\cdot\vec{\nabla}\left(\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right)\right]\vec{\Omega}\,\mathrm{d}^2\Omega}_{I_b}\end{aligned} \tag{2.122}$$

$$I_a=\frac{1}{4\pi}\int_{4\pi}\left[\vec{\Omega}\cdot\vec{\nabla}\left(\phi(\vec{r},E)\right)\right]\vec{\Omega}\,\mathrm{d}^2\Omega \tag{2.123}$$

$$=\frac{1}{4\pi}\frac{4\pi}{3}\vec{\nabla}\phi(\vec{r},E) \tag{2.124}$$

$$=\frac{1}{3}\overrightarrow{\mathrm{grad}}\phi(\vec{r},E) \tag{2.125}$$

For I_b we have to work on the integrand: we expand the gradient:

$$\vec{\nabla}\left(\vec{\Omega}\cdot\vec{J}\right)=\left(\vec{\Omega}\cdot\vec{J}\right)\vec{J}+\left(\vec{J}\cdot\vec{\nabla}\right)\vec{\Omega}+\vec{\Omega}\wedge\left(\vec{\nabla}\wedge\vec{J}\right)+\vec{J}\wedge\left(\vec{\nabla}\wedge\vec{\Omega}\right). \tag{2.126}$$

We take the scalar product with $\vec{\Omega}$:

$$\vec{\nabla}\left(\vec{\Omega}\cdot\vec{J}\right)\cdot\vec{\Omega}=\left(\vec{\Omega}\cdot\vec{J}\right)\vec{J}\cdot\vec{\Omega}+\left(\vec{J}\cdot\vec{\nabla}\right)\underbrace{\vec{\Omega}\cdot\vec{\Omega}}_{=1}+\underbrace{\vec{\Omega}\wedge\left(\vec{\nabla}\wedge\vec{J}\right)\cdot\vec{\Omega}}_{=0\text{ because }\vec{\Omega}\perp\vec{\Omega}\wedge(\cdots)}+\vec{J}\wedge\underbrace{\left(\vec{\nabla}\wedge\vec{\Omega}\right)}_{=\overrightarrow{\mathrm{rot}}\,\vec{\Omega}=\vec{0}}\cdot\vec{\Omega} \tag{2.127}$$

$$=\left(\vec{\Omega}\cdot\vec{J}\right)\vec{J}\cdot\vec{\Omega}+\vec{J}\cdot\vec{\nabla}. \tag{2.128}$$

Recalling that $\overrightarrow{\mathrm{rot}}\,\vec{\Omega}=\vec{0}$ because $\vec{\Omega}$ is a unit vector, we replace in the integral I_b:

$$I_b=\frac{3}{4\pi}\int_{4\pi}\left[\vec{\Omega}\cdot\vec{\nabla}\left(\vec{\Omega}\cdot\vec{J}(\vec{r},E)\right)\right]\vec{\Omega}\,\mathrm{d}^2\Omega \tag{2.129}$$

$$=\frac{3}{4\pi}\underbrace{\int_{4\pi}\left(\vec{\Omega}\cdot\vec{J}\right)\left(\vec{J}\cdot\vec{\Omega}\right)\vec{\Omega}\,\mathrm{d}^2\Omega}_{=\vec{0}}+\frac{3}{4\pi}\underbrace{\int_{4\pi}\left(\vec{J}\cdot\vec{\nabla}\right)\vec{\Omega}\,\mathrm{d}^2\Omega}_{=\vec{0}} \tag{2.130}$$

$$=\vec{0}. \tag{2.131}$$

Thus, we have:

$$I_1=\int_{4\pi}\left(\vec{\Omega}\cdot\vec{\nabla}\left[\phi(\vec{r},E,\vec{\Omega})\right]\right)\vec{\Omega}\,\mathrm{d}^2\Omega$$

$$=\frac{1}{3}\overrightarrow{\mathrm{grad}}\phi(\vec{r},E). \tag{2.132}$$

$$I_2=\int_{4\pi}\Sigma(\vec{r},E)\,\phi(\vec{r},E,\vec{\Omega})\,\vec{\Omega}\,\mathrm{d}^2\Omega \tag{2.133}$$

$$=\Sigma(\vec{r},E)\vec{J}(\vec{r},E)\text{ by definition of }\vec{J}(\vec{r},E). \tag{2.134}$$

$$
\begin{aligned}
I_3 &= \int_{4\pi}\int_{4\pi}\int_0^{+\infty} \Sigma_{\mathrm{s}}(\vec{r}, E \leftarrow E', \vec{\Omega} \leftarrow \vec{\Omega}')\, \phi(\vec{r}, E', \vec{\Omega}')\, \vec{\Omega}\, \mathrm{d}E'\, \mathrm{d}^2\Omega'\, \mathrm{d}^2\Omega \\
&= \int_{4\pi}\int_{4\pi}\int_0^{+\infty} \left[\frac{1}{4\pi}\Sigma_{\mathrm{s}0}(\vec{r}, E \leftarrow E') + \frac{3}{4\pi}\Sigma_{\mathrm{s}1}(E \leftarrow E')\, \vec{\Omega}\cdot\vec{\Omega}')\right] \\
&\qquad\qquad \times \left[\frac{1}{4\pi}\phi(\vec{r}, E') + \frac{3}{4\pi}\vec{\Omega}'\cdot\vec{J}(\vec{r}, E')\right] \vec{\Omega}\, \mathrm{d}E'\, \mathrm{d}^2\Omega'\, \mathrm{d}^2\Omega \\
&= \left(\frac{3}{4\pi}\right)^2 \int_{4\pi}\int_{4\pi}\int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E')\, \left(\vec{\Omega}\cdot\vec{\Omega}'\right)\, \left(\vec{\Omega}'\cdot\vec{J}(\vec{r}, E')\right)\, \vec{\Omega}\, \mathrm{d}E'\, \mathrm{d}^2\Omega'\, \mathrm{d}^2\Omega \qquad (2.135)
\end{aligned}
$$

Other terms are null because $\displaystyle\int_{4\pi} \vec{\Omega}\, \mathrm{d}^2\Omega = \vec{0}$ and $\displaystyle\int_{4\pi} \vec{\Omega}'\, \mathrm{d}^2\Omega' = \vec{0}$

$$
\begin{aligned}
I_3 &= \left(\frac{3}{4\pi}\right)^2 \int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E') \int_{4\pi} \left(\vec{\Omega}'\cdot\vec{J}(\vec{r}, E')\right) \underbrace{\int_{4\pi} \left(\vec{\Omega}\cdot\vec{\Omega}'\right)\, \vec{\Omega}\, \mathrm{d}^2\Omega}_{=\frac{4\pi}{3}\vec{\Omega}'}\, \mathrm{d}^2\Omega'\, \mathrm{d}E' \\
&= \frac{3}{4\pi}\int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E') \underbrace{\int_{4\pi} \left(\vec{\Omega}'\cdot\vec{J}(\vec{r}, E')\right)\, \vec{\Omega}'\, \mathrm{d}^2\Omega'}_{=\frac{4\pi}{3}\vec{J}(\vec{r}, E')}\, \mathrm{d}E' \\
&= \int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E')\, \vec{J}(\vec{r}, E')\, \mathrm{d}E' \qquad (2.136)
\end{aligned}
$$

The fission source disappears:

$$
\begin{aligned}
I_4 &= \frac{1}{4\pi K_{\mathrm{eff}}} \sum_{j=1}^{J^{\mathrm{fiss}}} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r}, E') \int_{4\pi} \phi(\vec{r}, E', \vec{\Omega}') \int_{4\pi} \vec{\Omega}\, \mathrm{d}^2\Omega\, \mathrm{d}^2\Omega'\, \mathrm{d}E' \qquad (2.137) \\
&= \frac{1}{4\pi K_{\mathrm{eff}}} \sum_{j=1}^{J^{\mathrm{fiss}}} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r}, E') \Bigg\{ \frac{1}{4\pi}\phi(\vec{r}, E') \underbrace{\int_{4\pi} \vec{\Omega}'\, \mathrm{d}^2\Omega'}_{=\vec{0}} \qquad (2.138) \\
&\qquad\qquad + \frac{3}{4\pi} \underbrace{\int_{4\pi} \left(\vec{\Omega}'\cdot\vec{J}(\vec{r}, E')\right)\, \mathrm{d}^2\Omega'}_{=\vec{0}} \Bigg\} \underbrace{\int_{4\pi} \vec{\Omega}\, \mathrm{d}^2\Omega}_{=\vec{0}}\, \mathrm{d}E' \qquad (2.139) \\
&= \vec{0}. \qquad (2.140)
\end{aligned}
$$

Finally, we obtain the second equation:

$$
\frac{1}{3}\overrightarrow{\mathrm{grad}}\, \phi(\vec{r}, E) + \Sigma(\vec{r}, E)\vec{J}(\vec{r}, E) = \int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E')\, \vec{J}(\vec{r}, E')\, \mathrm{d}E' \qquad (2.141)
$$

The P_1 approximation consists in resolving the system [8]:

$$\begin{cases} \mathrm{div}\vec{J}(\vec{r},E) + \Sigma\phi(\vec{r},E) = \displaystyle\int_0^{+\infty} \Sigma_{\mathrm{s}0}(E \leftarrow E')\phi(\vec{r},E')\,\mathrm{d}E' \\ \qquad\qquad\qquad\qquad + \dfrac{1}{K_{\mathrm{eff}}} \displaystyle\sum_{j=1}^{J^{\mathrm{fiss}}} \chi_j(E) \int_0^{+\infty} \nu\Sigma_{\mathrm{f},j}(\vec{r},E')\phi(\vec{r},E')\,\mathrm{d}E' \\ \dfrac{1}{3}\overrightarrow{\mathrm{grad}}\,\phi(\vec{r},E) + \Sigma(\vec{r},E)\vec{J}(\vec{r},E) = \displaystyle\int_0^{+\infty} \Sigma_{\mathrm{s}1}(E \leftarrow E')\,\vec{J}(\vec{r},E')\,\mathrm{d}E' \end{cases}$$

2.12.2 The equivalence of P_1 and Diffusion in one-group theory

We will now assume that all neutrons have the same energy: the P_1 system becomes [8]:

$$\begin{cases} \mathrm{div}\vec{J}(\vec{r}) + \Sigma\phi(\vec{r}) = \Sigma_{\mathrm{s}0}\phi(\vec{r}) + \underbrace{\dfrac{1}{K_{\mathrm{eff}}} \displaystyle\sum_{j=1}^{J^{\mathrm{fiss}}} \nu\Sigma_{\mathrm{f},j}(\vec{r})\phi(\vec{r}))}_{=S(\vec{r}} \\ \dfrac{1}{3}\overrightarrow{\mathrm{grad}}\,\phi(\vec{r}) + \Sigma(\vec{r})\vec{J}(\vec{r}) = \Sigma_{\mathrm{s}1}\,\vec{J}(\vec{r}) \end{cases}$$

Notice that the macroscopic differential cross-section $\Sigma_{\mathrm{s}1}(E \leftarrow E')$ had the dimension of a macroscopic cross-section divided by energy ($cm^{-1}.J^{-1}$) whereas $\Sigma_{\mathrm{s}1}$ is a macroscopic cross-section, whose dimension is cm^{-1}. Therefore, the previous formulae are homogeneous.

We can rewrite those equations with the average cosine of the scattering angle $\bar{\mu}_0 = \frac{2}{3A}$ and the transport cross-section Σ_{tr}:

$$\begin{aligned} &\bar{\mu}_0 = \frac{\Sigma_{\mathrm{s}1}}{\Sigma_{\mathrm{s}0}} \qquad \Sigma_{\mathrm{tr}} = \Sigma_{\mathrm{s}0}(1-\bar{\mu}_0) \quad \Sigma_a = \Sigma - \Sigma_{\mathrm{s}0} \\ &\Rightarrow \begin{cases} \mathrm{div}\vec{J}(\vec{r}) + \Sigma_{\mathrm{a}}\phi(\vec{r}) = S(\vec{r}) \\ \dfrac{1}{3}\overrightarrow{\mathrm{grad}}\,\phi(\vec{r}) + \left[\Sigma_{\mathrm{a}}(\vec{r}) + \Sigma_{\mathrm{tr}}(\vec{r})\right]\vec{J}(\vec{r}) = 0. \end{cases} \end{aligned} \tag{2.142}$$

In the second equation, we recognize Fick's law:

$$\vec{J}(\vec{r}) = -D\,\overrightarrow{\mathrm{grad}}\,\phi(\vec{r}) \text{ with } D = \frac{1}{3(\Sigma_{\mathrm{tr}} + \Sigma_{\mathrm{a}})} \tag{2.143}$$

The substitution into 2.142 leads to a neutron diffusion equation [8]:

$$\begin{aligned} & D\Delta\phi(\vec{r}) - \Sigma_{\mathrm{a}}\phi(\vec{r}) + S(\vec{r}) = 0. \\ \Leftrightarrow \quad & \Delta\phi(\vec{r}) - \frac{1}{L^2}\phi(\vec{r}) + \frac{S(\vec{r})}{D} = 0 \text{ with } L = \sqrt{\frac{D}{\Sigma_{\mathrm{a}}}} = \sqrt{\frac{1}{3\Sigma_{\mathrm{a}}(\Sigma_{\mathrm{tr}} + \Sigma_{\mathrm{a}})}} \end{aligned} \tag{2.144}$$

When we just have one energy group [8], the previous development demonstrates that solving the P_1 equation *even with a linear anisotropic source* is equivalent to solving the diffusion equation. In this case, we could mimic a diffusion calculation with a P_1 solver by performing the following change in the solver:

$$\tilde{\Sigma}_{\text{total}} = \frac{1}{3D} = \Sigma_{\text{a}} + \Sigma_{\text{tr}} \tag{2.145}$$

$$\tilde{\Sigma}_{\text{s1}} = 0 \tag{2.146}$$

$$\tilde{\Sigma}_{\text{s0}} = \Sigma_{\text{tr}} = \frac{1}{3D} - \Sigma_{\text{a}} \tag{2.147}$$

assuming that we know D, Σ_{total} (or Σ_{a}) and Σ_{tr}.

Replacing the normal cross-sections in 2.142 by the cross-sections defined with a tilde above will lead to the exact formulation of the diffusion equation using the P_1 solver. It is therefore a good way to mimic a diffusion code because the two solvers should give exactly the same answer [1].

The independence of cross-sections and fluxes from energy is obviously wrong in the reality. We will then see what happens when we consider more than one group of energy.

2.12.3 Equivalence of P_1 equations and diffusion theory with a multi-group formulation: the problem of anisotropy

With more than one group, we cannot derive the Fick law as done previously because the second equation for the P_1 system is written:

$$\frac{1}{3}\overrightarrow{\text{grad}}\,\phi(\vec{r},E) + \Sigma(\vec{r},E)\vec{J}(\vec{r},E) = \int_0^{+\infty} \Sigma_{\text{s1}}(E \leftarrow E')\,\vec{J}(\vec{r},E')\,\text{d}E' \tag{2.148}$$

In this equation, we see that $\phi(\vec{r},E)$ is not only related to $\vec{J}(\vec{r},E)$, but also to all other currents at other energies through the scattering term. The multi-group formulation of the P_1 equations becomes [8]:

$$\forall g \in [\![1,G]\!],\ \begin{cases} \text{div}\vec{J}_g(\vec{r}) + \Sigma_g^0\phi_g(\vec{r}) = \displaystyle\sum_{h=1}^{G} \Sigma_{\text{s0},g\leftarrow h}\phi_h(\vec{r}) \\ \qquad\qquad + \dfrac{1}{K_{\text{eff}}} \displaystyle\sum_{j=1}^{J^{\text{fiss}}} \chi_{g,j} \sum_{h=1}^{G} \nu_h \Sigma_{\text{f},j,h}(\vec{r})\phi_h(\vec{r}) \\ \dfrac{1}{3}\overrightarrow{\text{grad}}\,\phi_g(\vec{r}) + \Sigma_g^1(\vec{r})\vec{J}_g(\vec{r}) = \displaystyle\sum_{h=1}^{G} \Sigma_{\text{s1},g\leftarrow h}\,\vec{J}_h(\vec{r}) \end{cases}$$

Note that Σ_g^0 is weighted by the flux whereas Σ_g^1 is weighted by the current.

We rewrite those equations in a matrix form: we define the vector of the flux, carrying all fluxes in G groups and the matrix of the current carrying the three coordinates of the current vectors of the G groups:

$$\vec{\Phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_G \end{pmatrix} \qquad \underline{\underline{J}} = \left(J_{g,j} \right)_{\substack{g \in [\![1,G]\!] \\ j \in [\![1,3]\!]}} = \begin{pmatrix} J_{1,1} & J_{1,2} & J_{1,3} \\ J_{2,1} & J_{2,2} & J_{2,3} \\ \vdots & \vdots & \vdots \\ J_{G,1} & J_{G,2} & J_{G,3} \end{pmatrix} \tag{2.149}$$

Then the system of P_1 equations becomes:

$$\begin{cases} \overrightarrow{\mathrm{Div}} \underline{\underline{J}}(\vec{r}) + \underline{\underline{\Sigma}}^0 \vec{\Phi}(\vec{r}) = \underline{\underline{\Sigma}}_{\mathrm{s0}} \vec{\Phi}(\vec{r}) + \dfrac{1}{K_{\mathrm{eff}}} \underline{\underline{\Sigma}}_{\mathrm{f}}(\vec{r}) \vec{\Phi}(\vec{r}) \\ \dfrac{1}{3} \underline{\underline{\mathrm{Grad}}}\, \vec{\Phi}(\vec{r}) + \underline{\underline{\Sigma}}^1(\vec{r}) \underline{\underline{J}}(\vec{r}) = \underline{\underline{\Sigma}}_{\mathrm{s1}}\, \underline{\underline{J}}(\vec{r}) \end{cases} \tag{2.150}$$

It makes sense with following definitions:

$$\begin{aligned} \overrightarrow{\mathrm{Div}} \underline{\underline{J}} &= \left(\textstyle\sum_{j=1}^{3} \frac{\partial J_{i,j}}{\partial x_j} \right)_{i \in [\![1,G]\!]} = \begin{pmatrix} \frac{\partial J_{1,1}}{\partial x_1} + \frac{\partial J_{1,2}}{\partial x_2} + \frac{\partial J_{1,3}}{\partial x_3} \\ \frac{\partial J_{2,1}}{\partial x_1} + \frac{\partial J_{2,2}}{\partial x_2} + \frac{\partial J_{2,3}}{\partial x_3} \\ \vdots \\ \frac{\partial J_{G,1}}{\partial x_1} + \frac{\partial J_{G,2}}{\partial x_2} + \frac{\partial J_{G,3}}{\partial x_3} \end{pmatrix} \\ &= \left[\begin{pmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} J_{1,1} & J_{2,1} & J_{3,1} & \dots & J_{G,1} \\ J_{2,1} & J_{2,2} & J_{3,2} & \dots & J_{G,2} \\ J_{3,1} & J_{2,3} & J_{3,3} & \dots & J_{G,3} \end{pmatrix}}_{\underline{\underline{J}}^T} \right]^T \end{aligned} \tag{2.151}$$

$$\underline{\underline{\Sigma}}_{\mathrm{s0}} = \left(\Sigma_{\mathrm{s0},g \leftarrow j} \right)_{\substack{g \in [\![1,G]\!] \\ j \in [\![1,G]\!]}} = \begin{pmatrix} \Sigma_{\mathrm{s0},1\leftarrow 1} & \Sigma_{\mathrm{s0},1\leftarrow 2} & \dots & \Sigma_{\mathrm{s0},1\leftarrow G} \\ \Sigma_{\mathrm{s0},2\leftarrow 1} & \Sigma_{\mathrm{s0},2\leftarrow 2} & \dots & \Sigma_{\mathrm{s0},2\leftarrow G} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathrm{s0},G\leftarrow 1} & \Sigma_{\mathrm{s0},G\leftarrow 2} & \dots & \Sigma_{\mathrm{s0},G\leftarrow G} \end{pmatrix} \tag{2.152}$$

$$\underline{\underline{\Sigma}}^0 = \mathrm{diag}\left(\Sigma_g^0\right)_{g\in[\![1,G]\!]} = \begin{pmatrix} \Sigma_1^0 & 0 & \dots & 0 \\ 0 & \Sigma_2^0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_G^0 \end{pmatrix} \tag{2.153}$$

$$\underline{\underline{\Sigma}}_{\mathrm{f}} = \left(\nu_j\,\chi_{j,g}\Sigma_{\mathrm{f}j,g}\right)_{\substack{j\in[\![1,J^{\mathrm{fiss}}]\!] \\ g\in[\![1,G]\!]}} = \begin{pmatrix} \nu_1\,\chi_{1,1}\Sigma_{\mathrm{f}1,1} & \nu_1\,\chi_{1,2}\Sigma_{\mathrm{f}1,2} & \dots & \nu_1\,\chi_{1,G}\Sigma_{\mathrm{f}1,G} \\ \nu_2\,\chi_{2,1}\Sigma_{\mathrm{f}2,1} & \nu_2\,\chi_{2,2}\Sigma_{\mathrm{f}2,2} & \dots & \nu_2\,\chi_{2,G}\Sigma_{\mathrm{f}2,G} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{J^{\mathrm{fiss}}}\,\chi_{J^{\mathrm{fiss}},1}\Sigma_{\mathrm{f}J^{\mathrm{fiss}},1} & \nu_{J^{\mathrm{fiss}}}\,\chi_{J^{\mathrm{fiss}},2}\Sigma_{\mathrm{f}J^{\mathrm{fiss}},2} & \dots & \nu_{J^{\mathrm{fiss}}}\,\chi_{J^{\mathrm{fiss}},G}\Sigma_{\mathrm{f}J^{\mathrm{fiss}},G} \end{pmatrix}$$

And for the second equation:

$$\underline{\underline{\mathrm{Grad}}}\,\vec{\Phi} = \left(\frac{\partial\phi_g}{\partial x_j}\right)_{\substack{g\in[\![1,G]\!] \\ j\in[\![1,3]\!]}} = \begin{pmatrix} \frac{\partial\phi_1}{\partial x_1} & \frac{\partial\phi_1}{\partial x_2} & \frac{\partial\phi_1}{\partial x_3} \\ \frac{\partial\phi_2}{\partial x_1} & \frac{\partial\phi_2}{\partial x_2} & \frac{\partial\phi_2}{\partial x_3} \\ \vdots & \vdots & \vdots \\ \frac{\partial\phi_G}{\partial x_1} & \frac{\partial\phi_G}{\partial x_2} & \frac{\partial\phi_G}{\partial x_3} \end{pmatrix} = \left[\begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_G \end{pmatrix}}_{\vec{\Phi}^T} \right]^T \tag{2.154}$$

$$\underline{\underline{\Sigma}}^1 = \mathrm{diag}\left(\Sigma_g^1\right)_{g\in[\![1,G]\!]} = \begin{pmatrix} \Sigma_1^1 & 0 & \dots & 0 \\ 0 & \Sigma_2^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_G^1 \end{pmatrix} \tag{2.155}$$

$$\underline{\underline{\Sigma}}_{\mathrm{s1}} = \left(\Sigma_{\mathrm{s1},g\leftarrow j}\right)_{(g,j)\in[\![1,G]\!]^2} = \begin{pmatrix} \Sigma_{\mathrm{s1},1\leftarrow 1} & \Sigma_{\mathrm{s1},1\leftarrow 2} & \dots & \Sigma_{\mathrm{s1},1\leftarrow G} \\ \Sigma_{\mathrm{s1},2\leftarrow 1} & \Sigma_{\mathrm{s1},2\leftarrow 2} & \dots & \Sigma_{\mathrm{s1},2\leftarrow G} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathrm{s1},G\leftarrow 1} & \Sigma_{\mathrm{s1},G\leftarrow 2} & \dots & \Sigma_{\mathrm{s1},G\leftarrow G} \end{pmatrix} \tag{2.156}$$

So that we have, for the scattering terms for example, the matrix:

$$\begin{aligned}\underline{\underline{\Sigma}}_{\mathrm{s1}}\underline{\underline{J}} &= \left(\sum_{k=1}^{G}\Sigma_{\mathrm{s1},g\leftarrow k}J_{k,j}\right)_{\substack{g\in[\![1,G]\!]\\ j\in[\![1,3]\!]}} \\ &= \begin{pmatrix} \sum_{k=1}^{G}\Sigma_{\mathrm{s1},1\leftarrow k}J_{k,1} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},2\leftarrow k}J_{k,2} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},G\leftarrow k}J_{k,3} \\ \sum_{k=1}^{G}\Sigma_{\mathrm{s1},2\leftarrow k}J_{k,1} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},2\leftarrow k}J_{k,2} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},2\leftarrow k}J_{k,3} \\ \vdots & \vdots & \vdots \\ \sum_{k=1}^{G}\Sigma_{\mathrm{s1},G\leftarrow k}J_{k,1} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},G\leftarrow k}J_{k,2} & \sum_{k=1}^{G}\Sigma_{\mathrm{s1},G\leftarrow k}J_{k,3} \end{pmatrix}. \end{aligned} \tag{2.157}$$

We will then try to formulate the second equation in terms of Fick's law:

$$\begin{aligned} & \frac{1}{3}\underline{\underline{\mathrm{Grad}}}\,\vec{\Phi}(\vec{r}) + \underline{\underline{\Sigma}}^{1}(\vec{r})\underline{\underline{J}}(\vec{r}) = \underline{\underline{\Sigma}}_{\mathrm{s1}}(\vec{r})\,\underline{\underline{J}}(\vec{r}) \\ \Leftrightarrow \quad & 3\left(\underline{\underline{\Sigma}}^{1}(\vec{r}) - \underline{\underline{\Sigma}}_{\mathrm{s1}}(\vec{r})\right)\underline{\underline{J}}(\vec{r}) = -\underline{\underline{\mathrm{Grad}}}\,\vec{\Phi}(\vec{r}) \end{aligned} \tag{2.158}$$

We can suppose that the matrix in front of $\underline{\underline{J}}(\vec{r})$ is invertible, so that:

$$\exists\,\underline{\underline{D}} \in M_n(\mathbb{R})/ \quad \underline{\underline{D}} = \frac{1}{3}\left(\underline{\underline{\Sigma}}^{1}(\vec{r}) - \underline{\underline{\Sigma}}_{\mathrm{s1}}(\vec{r})\right)^{-1} \tag{2.159}$$

We get:

$$\underline{\underline{D}}^{-1} = 3 \times \begin{pmatrix} \Sigma_1^1 - \Sigma_{\mathrm{s1},1\leftarrow 1} & -\Sigma_{\mathrm{s1},1\leftarrow 2} & \dots & -\Sigma_{\mathrm{s1},1\leftarrow G} \\ -\Sigma_{\mathrm{s1},2\leftarrow 1} & \Sigma_2^1 - \Sigma_{\mathrm{s1},2\leftarrow 2} & \dots & -\Sigma_{\mathrm{s1},2\leftarrow G} \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{\mathrm{s1},G\leftarrow 1} & -\Sigma_{\mathrm{s1},G\leftarrow 2} & \dots & \Sigma_G^1 - \Sigma_{\mathrm{s1},G\leftarrow G} \end{pmatrix} \tag{2.160}$$

A classic diffusion law implies that $\underline{\underline{D}}$ is diagonal, and therefore $\underline{\underline{D}}^{-1}$ is as well. This case is achieved *if and only if we consider isotropic scattering* because then $\underline{\underline{\Sigma_{\mathrm{s1}}}} = \underline{\underline{0}}$. Otherwise, there is no equivalence between the P_1 method and the diffusion method. Indeed, we cannot be sure that the diffusion coefficients D_g of each group are equal to $\frac{1}{3}(\Sigma_g^1 - \Sigma_{\mathrm{s1},g\leftarrow g})$, which would be equivalent to the assumption that the off-diagonal terms of the previous matrix $\underline{\underline{D}}^{-1}$ are equal to 0.

If we use the transport correction to calculate $\underline{\underline{\Sigma_{\mathrm{s1}}}}$, then we have:

$$\begin{aligned} & \Sigma_{\mathrm{s1},g\leftarrow h} = \bar{\mu}_0\Sigma_{\mathrm{s0},g\leftarrow h} \text{ with } \bar{\mu}_0 = \frac{2}{3A} \text{ constant;} \\ & \Sigma_{\mathrm{tr},g\leftarrow h} = (1-\bar{\mu}_0)\,\Sigma_{\mathrm{s0},g\leftarrow h}; \quad \Sigma_{\mathrm{r},g} = \Sigma_g - \Sigma_{\mathrm{s0},g\leftarrow g} \end{aligned} \tag{2.161}$$

$$\underline{\underline{D}}^{-1} = 3 \times \begin{pmatrix} \Sigma_{\mathrm{r},1} + \Sigma_{\mathrm{tr},1\leftarrow 1} & -\mu_0 \Sigma_{\mathrm{s0},1\leftarrow 2} & \dots & -\mu_0 \Sigma_{\mathrm{s0},1\leftarrow G} \\ -\mu_0 \Sigma_{\mathrm{s0},2\leftarrow 1} & \Sigma_{\mathrm{r},2} + \Sigma_{\mathrm{tr},2\leftarrow 2} & \dots & -\mu_0 \Sigma_{\mathrm{s0},2\leftarrow G} \\ \vdots & \vdots & \ddots & \vdots \\ -\mu_0 \Sigma_{\mathrm{s0},G\leftarrow 1} & -\mu_0 \Sigma_{\mathrm{s0},G\leftarrow 2} & \dots & \Sigma_{\mathrm{r},G} + \Sigma_{\mathrm{tr},G\leftarrow G} \end{pmatrix} \tag{2.162}$$

The problem remains the same: there is no formal equivalence with diffusion theory because $\underline{\underline{D}}$ is not diagonal. However, now there may be a way to diagonalize it because the linear anisotropic terms are proportional to the isotropic terms, so that we may arrive at Fick's law [8]. However this has to be shown. In addition, even if this were feasible, then the expression of the new matrix with the formalism of the first matrix may not be totally trivial.

A common procedure to try to mimic a diffusion code with a P_1 solver is described hereafter, as proposed in the INSTANT manual [1]:

$$\forall\, g \in [\![1, G]\!] : \tag{2.163}$$

$$\tilde{\Sigma}_{\mathrm{total},g} = \frac{1}{3D_g} = \Sigma_{\mathrm{r},g} + \Sigma_{\mathrm{tr},g} \tag{2.164}$$

$$\tilde{\Sigma}_{\mathrm{s1},g\leftarrow h} = 0 \quad \forall\, h \in [\![1, G]\!] \text{ and } h \neq g \tag{2.165}$$

$$\tilde{\Sigma}_{\mathrm{s0},g\leftarrow g} = \Sigma_{\mathrm{tr},g} = \frac{1}{3D_g} - \Sigma_{\mathrm{r},g} \tag{2.166}$$

$$\tilde{\Sigma}_{\mathrm{s0},g\leftarrow h} = \Sigma_{\mathrm{s0},g\leftarrow h} \quad \forall\, h \in [\![1, G]\!] \text{ and } h \neq g. \tag{2.167}$$

with the assumption that we know the diffusion coefficients, the transport and absorption cross-sections in each group.

It leads to the computation of the following matrix in the P_1 code:

$$\underline{\underline{\tilde{D}}}^{-1} = 3 \times \begin{pmatrix} \Sigma_{\mathrm{r},1} + \Sigma_{\mathrm{tr},1} & 0 & \dots & 0 \\ 0 & \Sigma_{\mathrm{r},2} + \Sigma_{\mathrm{tr},2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_{\mathrm{r},G} + \Sigma_{\mathrm{tr},G} \end{pmatrix} \neq \underline{\underline{D}}^{-1} \; ! \tag{2.168}$$

Such an approximation will miss the off-diagonal terms by assuming them to be equal to 0, and therefore calculate a wrong matrix $\underline{\underline{D}}$. Therefore, *it will not be a pure diffusion calculation*, and it will also not be a correct P_1 calculation because:

- we do not consider pure isotropic scattering: the diagonal terms contain a part of the linear anisotropic scattering information through the transport cross-sections $\Sigma_{\mathrm{tr},g}$;
- we do not consider a true linear anisotropic scattering source with a transport correction to calculate those terms because we do not take into account the off-diagonal terms of

$\underline{\underline{\Sigma_{s1}}}$;

– we obviously do not consider a linear anisotropic scattering that would not be calculated through the transport correction.

In conclusion, there is no easy way to mimic a diffusion code with a P_1 solver in a multi-group situation *unless we consider pure isotropic scattering* or we find a way to analytically diagonalize the transport corrected $\underline{\underline{\Sigma_{s1}}}$ and then find the correct combination of terms to replace in the P_1 solver.

However, in the case of graphite, we may assume [8]:

– that the anisotropic terms are small compared to the isotropic terms. Indeed, the graphite atom is much heavier than the neutron, which tends to create nearly-isotropic scattering. Anisotropic terms of order higher than 1 may be easily neglected, and the transport correction probably gives a good representation of the linear anisotropy.

– that the off-diagonal terms of $\underline{\underline{\Sigma_{s1}}}$ are small compared to the diagonal terms. This may be especially true with fewer energy groups because the graphite is not a very good moderator [7]. It means that after a collision, the loss of lethargy of the neutron is small so that the neutrons have more chances to stay in the same energy group than to switch to another group. The fewer groups we have, the larger they are, so that this approximation becomes better.

In this case, we can:

– keep an anisotropy of order 1 for the neutron source using the transport correction;

– and neglect the off-diagonal terms of $\underline{\underline{\Sigma_{s1}}}$ in our calculation.

With this set of assumptions, we arrive at Fick's law and we can use the equivalence from equation 2.167 to mimic the results of a diffusion code [1].

The other choice is to let the code perform the normal P_1 calculation with a linear anisotropic source provided by the $\underline{\underline{\Sigma_{s1}}}$ evaluation data processed in DRAGON. This solution was finally chosen in the thesis as shown later.

2.12.4 The P_3 equations

The development in spherical harmonics of the flux can be extended to higher orders. In INSTANT, the P_3 method was used which means that we calculated the flux with an approximation of order 3. The scattering source may have a different expansion. For example, a linear anisotropic source (order 1) was used. In this case, the transport correction was not used and the $\underline{\underline{\Sigma_{s1}}}$ matrices given by DRAGON were preferred. Those matrices come directly from the evaluation. DRAGON is only producing homogenized, condensed matrices using the flux calculated on the lattice level. This procedure may lead to better results than the P_1 because a higher order both for the source and the flux calculation is being used.

On the contrary, for example, it is not trivial to predict which calculation would be the best between a P_1, transport corrected solution and a P_3 solution with an isotropic source.

2.13 The Monte-Carlo method: a different philosophy

The Monte-Carlo method uses a completely different philosophy. The idea is to simulate the random walk of a neutron from its creation (by fission or (n,xn) reactions) to its death (by absorption or leakage) in the core [6]. Records are kept of every reaction that the neutron encounters during its life. Millions of simulations enable one to produce reaction rates with statistical meaning. The statistical uncertainty decreases proportional to $\frac{1}{\sqrt{N}}$ where N is the number of neutrons simulated. Therefore, the convergence is slow, and such a method requires a great deal of computation time. However, this method is a perfect candidate for parallelization because in theory we can assign one neutron per processor: as stated at the beginning of the section, we neglect the neutron-neutron interactions.

The Monte-Carlo method has many advantages. For example, any geometry can be defined and we can use a continuous energy structure so that the self-shielding step disappears from the calculation. The method is meant to be exact, as far as the cross-sections are correct and the number of simulated particles is sufficient. Thus, the Monte-Carlo codes are often used to validate deterministic codes. The method is also used in other applications like detector studies, criticality studies, medical studies, fusion studies, and especially when experimental results are not available. It faces its own problems, like for example statistical instabilities.

Several codes have been developed to implement the Monte-Carlo method for reactor physics problems. MCNP is a well-known one, developed at the Los Alamos National Laboratory. Its 5^{th} version served as a reference for this study [19].

SERPENT is another Monte-Carlo code developed by Jaako Leppänen for VTT (Valtion Teknillinen Tutkimuskeskus, Technical Research Centre of Finland) which was designed for the generation of cross-sections [20]. It is the first example of a Monte-Carlo code which has been built for operational purposes. Usually calculations related to design or reactor operations are carried out with deterministic codes. With the increase of calculation speed and massive parallelization, stochastic production codes like SERPENT may become the future of reactor analysis.

2.14 Isotopic depletion

When we operate a reactor, the material inside the core is changing due to two phenomena [6]:

– the nuclear reactions resulting from collisions between neutrons and nuclei;

– the radioactive decay of some nuclei.

In the case of the nuclear reactions produced by exposure to neutron flux, the variation of the number density is proportional to the reaction rate, that is to say the probability to have an absorption reaction:

$$\frac{\mathrm{d}N(t)}{\mathrm{d}t} = -\langle \sigma_{\mathrm{a}} \phi(t) \rangle \, N(t) \tag{2.169}$$

with

$$\langle \sigma_{\mathrm{a}} \phi(t) \rangle = \int_0^{+\infty} \sigma_{\mathrm{a}}(u) \phi(u,t) \, \mathrm{d}u. \tag{2.170}$$

This equation does not depend on which mode of decay the compound nucleus will follow: it can be fission, radiative capture, etc. The only important information is that we know that this reaction will not produce the initial nucleus. All reactions are taken into account except the elastic and inelastic scattering reactions:

$$\sigma_{\mathrm{a}} = \sigma_{\mathrm{total}} - \sigma_{\mathrm{e}} - \sigma_{\mathrm{in}}. \tag{2.171}$$

In the case of the radioactive decay of an isotope, the number density of a radioactive nucleus obeys to a simple differential equation:

$$\frac{\mathrm{d}N(t)}{\mathrm{d}t} = -\lambda N(t) \tag{2.172}$$

λ is called the radioactive decay constant and is independent of time, but varies depending on the way of decay and the isotope considered.

In a reactor, nuclear reactions created by the flux and radioactive decay occur together on all isotopes at the same time, and an isotope can be produced by reactions or decays involving other isotopes. Therefore we can write for each single nuclide k [8]:

$$\begin{aligned} \frac{\mathrm{d}N_k(t)}{\mathrm{d}t} = \phi(t) \sum_{i=1}^{M} N_i(t) \sigma_{\mathrm{f},i} y_{i,k} + \phi(t) \sum_{\substack{s=1 \\ s \neq k}}^{M} \sum_{x} N_s(t) \sigma_{x,s} + \phi(t) \sum_{j=1}^{M} N_j(t) \lambda_j \alpha_{j,k} \\ - \phi(t) \sum_{x} \sigma_{x,k} N_k(t) - \lambda_k N_k(t) \end{aligned} \tag{2.173}$$

where

N_k = number density of isotope k

$\phi(t)$ = flux

$\sigma_{\mathrm{f},i}$ = fission cross-section of isotope i

$\sigma_{x,s}$ = cross-section of a reaction of a neutron with isotope s (for example (n,γ)) that produces isotope k

λ_k = decay constant of isotope k

$y_{i,k}$ = yield of isotope k due to a fission of isotope i

$\alpha_{j,k}$ = probability that the decay of isotope j produces isotope k

A system is obtained containing an equation describing the evolution for each nuclide. Those equations are coupled between all nuclides. The information on decay channels and reaction yields is stored in the nuclear reaction chains and is available in the evaluations.

The difficult part of solving such a system is that the number of ways for a nuclide to evolve can be large, and it is highly coupled to the depletion of other nuclides. The time constants can also be very different, which may create numerical issues [6].

In addition, the depletion affects the flux. Indeed, macroscopic cross-sections have to be updated regarding the number density, but it also indirectly affects the reaction rates, because the spectrum may change according to the new core configuration. However, those changes are relatively slow. A common assumption is made and consists of discretizing the time into time-steps. The flux is then considered constant during each time step, and then is recalculated. We always stay within the steady-state approximation [6].

The calculation is usually performed at a constant specific power (power per mass of initial heavy isotopes). This power is used to scale the flux so that its magnitude acquires some sense. In this model, the power at the beginning and end of the time step is equal. It enables us to calculate the flux at the end of the time step. A few iterations are needed to correctly calculate the number densities at the end of the time step according to the specific power.

The burnup $B(t)$ measures a sort of average of the energy released by an assembly per initial mass of heavy isotopes [6]. It is usually expressed in MW day .tonne^{-1}. This value provides some information about the energy that has been extracted from an assembly during his life. Mathematically, we define it at time t_f by:

$$B(t) = \frac{V}{W} \int_0^{t_f} \langle H\phi(t') \rangle \, \mathrm{d}t'. \tag{2.174}$$

where V is the volume of the fuel and W is the weight of heavy isotopes at $t = 0$. H is a factor giving the recoverable energy from neutron-induced reactions.

Depletion calculations have to be performed at the lattice level to generate microscopic cross-sections for the core. The cross-sections are then tabulated with the burnup and passed to the core calculation. The number densities for the core may be recovered either from the lattice calculation or, less frequently, from the core calculation if the solver contains an in-core depletion module. The principle of the depletion remains the same in the lattice as in the core.

CHAPTER 3

DESCRIPTION OF THE DIFFERENT WAYS TO PREPARE CROSS-SECTIONS

3.1 General considerations on the calculation schemes

3.1.1 Description of the calculation schemes

As stated in chapter 2, a detailed full-core 3D transport calculation would require a huge amount of memory and would take significant CPU time to be used by utilities in production calculations or even in laboratories for design purposes. While trying to produce accurate cross-sections for this reactor, we should keep in mind that utilities need to perform numerous core calculations to predict the fuel depletion during approximately one year of operation, verify that reactivity margins are respected during the whole operation cycle, and analyse many core configurations. Design studies also require several runs of core calculation to establish the behaviour in transient conditions. Performing all of these studies with a detailed full-core calculation, even in 2D, is currently not feasible within a reasonable amount of time.

Therefore, the typical way to proceed is to use a two step calculation scheme [6]:

- a detailed calculation at the assembly level with reflective boundary conditions, which gives homogenized cross-sections for the assemblies, condensed to a certain number of groups; this step is called the lattice calculation.
- a second calculation at the core level with homogenized properties in each assembly and usually a small number of groups. We call it the full-core calculation or whole-core calculation.

This scheme is particularly effective in the case of light water reactors because the typical mean free path of thermal neutrons in water is equal to few millimetres. Therefore, modelling the assemblies with a small layer of water around them is very effective at capturing the main physical effects. The coupling across the full core is low, which justifies this kind of approach. The assemblies in the full core exist largely in an infinite lattice condition with the exception of those at the periphery.

For high temperature reactors, the mean free-path of neutrons is much longer, and therefore the decoupling between the block [1] level and the full-core level is less accurate. Facing this problem, we however keep a two-step calculation scheme because an alternative has not been well established. However, knowing that the spectrum will completely differ between

1. For high temperature reactors, the term “block” replaces “assembly”.

the lattice and the core calculation, especially in the blocks adjacent to the reflector, particular care must be taken to provide cross-sections to the core calculation. The purpose of this thesis is to determine which models can be used at the lattice level to obtain good accuracy at the core level. The lattice calculations will be performed using DRAGON, whereas INSTANT will be used for full-core calculations. The temperature remains set at 293,6 K, as the study of its influence is beyond the scope of the thesis.

At the lattice level, a single block of fuel may be quite representative for the central blocks located far from reflectors. However, the blocks located at the periphery are receiving a lot of thermal neutrons coming back from the reflector. For them, the single block infinite lattice model may not be relevant due to the neutron energy spectral modification created by the reflector's presence. To deal with this specific feature of this reactor, it is assumed that keeping a large number of groups for the INSTANT calculation would compensate for the poor quality of the microscopic cross-sections used to describe those blocks. With this point of view, we can still generate microscopic cross-sections from a single block model at the lattice level if sufficient energy groups are retained for the whole core calculation. It will be the first calculation scheme evaluated, named "single block path".

Secondly, it is proposed to treat the peripheral blocks separately with supercell models including one row of neighbouring blocks around the block of interest at the lattice level. The other central blocks of the core remain calculated with the previous path. This method forms the second calculation scheme and is called "supercell path". Our assumption is that one may be able to incorporate some of the effects due to the surrounding graphite blocks, and have a calculation in INSTANT based on more accurate homogenized cross-sections. In addition, a quite large number of groups may also correct the remaining errors of the cross-sections passed to INSTANT. In this thesis, we aim to quantify the accuracy of such a calculation compared to the other path.

A full-core transport calculation would, in theory, provide the best solution to prepare cross-sections for the core model as the assumptions on the decoupling of the fuel cells vanishes. However, this is not practical with currently-available tools. DRAGON faces memory limitations due to addresses coded on 32 bits, and even with the fastest solver (`SYBILT:`) and no discretization of the geometry, the calculation is cumbersome. As the use of `EXCELT:` and the discretization has proved to be necessary on supercells, and given the fact that it was not possible to do it in DRAGON, this idea was not pursued. Note also that in general, most deterministic lattice codes were not designed to do this kind of calculation and few studies have been done to determine the precision of the codes with large models like that.

Moreover, if a full-core transport calculation was feasible within a reasonable time, then

there would be no interest to generate homogenized cross-sections for another core solver as the detailed solution of the problem is already known.

Passing the homogenized block cross-sections from DRAGON to INSTANT is not trivial. INSTANT needs macroscopic cross-sections for each block. DRAGON is able to produce either microscopic or macroscopic homogenized cross-sections. However, the macroscopic cross-sections are generated using the volumes upon which the homogenization is done. Therefore, if block sizes are different between the two codes, then the homogenized densities of atoms will differ between them, and the macroscopic homogenized cross-sections computed by DRAGON will be not be suitable for INSTANT. As it happens, this is the case in our models because of the actual limitation of the tracking modules in DRAGON. At present, it is impossible to model some hexagons embedded into a larger one with a straight boundary [3]. Therefore, we have to produce homogenized *microscopic* cross-sections in DRAGON and then reconstruct the *macroscopic* cross-sections for INSTANT using the correct number densities.

More details to circumvent this difficulty are given in appendices C, D, and in Section 3.4 when we detail the single block model built in DRAGON. Notice that this point is of great importance. The block volumes between DRAGON and INSTANT differ by 8,26 %, which leads to an error of the same amount on the value of all isotope's densities if we do not take care about it.

A program written in Fortran90 by Javier Ortensi (INL) enables us to recover on the one hand the homogenized microscopic cross-sections of every isotope used in a DRAGON calculation, and on the other hand the corresponding densities provided by the user. The program then calculates the correct homogenized macroscopic cross-sections for INSTANT and creates this part of the INSTANT input file. The calculation of the correct homogenized densities in INSTANT has been performed and is given in Appendices C and D. The difficult part of this calculation is to correctly deal with the volume packing fraction and the spherical geometry of the TRISO particles projected on a plane surface. The reader is encouraged to examine the details of this calculation in order to precisely understand the underlying problem.

It was decided to use the JEFF-3.1.0 evaluation with the SHEM-295 group-structure in DRAGON for all our calculations. The SHEM group structure was developed by Alain Hébert as an adaptation of the 281-group Santamarina-Hfaiedh energy mesh [21, 22] and of the 361-group energy mesh [23]. Its boundaries can be found in [13] and it is available at: `http://www.polymtl.ca/merlin/libraries.htm`.

3.1.2 Reference calculations: MCNP

MCNP models have been provided for every type of calculations by Michael Pope (INL). It is assumed that those models represent the best solutions at every step, considering the fact that no experimental data is available for this particular reactor.

Nevertheless, the double heterogeneity treatment for spherical TRISO particles in MCNP is questionable. It must be assembled from a 3-D lattice sized such that the correct packing fraction is achieved. Several possibilities may be used to create the lattice geometry of TRISO particles inside the pellet: squares, hexagons, hexagonal lattice with the corners being removed, etc. Even if we manage to get the same packing fraction, the geometrical differences introduce an uncertainty of about 100 pcm on the K_{eff}. This should be kept in mind in comparison of DRAGON to MCNP calculations.

At the lattice level, the multiplication factor (K_{eff}) obtained in MCNP and DRAGON will be compared. The error on the K_{eff} was the criterion to decide which models must be retained. This was used in choosing between the different tracking options in DRAGON.

At the core level, not only were the K_{eff} obtained in INSTANT and MCNP compared, but also the block fission rates. Indeed, this parameter is of very high interest because it will directly influence the distribution of power used for thermofluid calculations and also the flux used for depletion calculations. A good prediction of the fission rate shape is therefore essential, perhaps more than the K_{eff} prediction since in reality, this parameter is biased to meet the experimental results for each reactor.

A comparison of the detailed shapes of the flux or of the fission rates obtained with the two codes would also have been interesting, but MCNP is not able to produce this information easily. Our hope is that a good agreement between the block fission rates may indicate that the detailed fission rates are not too far from each other. In order to give some basis for this assumption, we will draw the shape of the fission rates across the core along one radius and compare this to MCNP. Results have been provided for every calculation path.

Notice here that the calculation in MCNP has been performed with continuous energy cross-sections and with much more detailed reactions than in our deterministic calculations. Thus, a true comparison with the same library is not possible. The same cross-section evaluation between DRAGON and MCNP was used: JEFF-3.1.0. Unfortunately, the thermal scattering cross-sections for the graphite from this evaluation were not available for MCNP at the laboratory. We therefore used the ENDFB-VII evaluation for those cross-sections. This difference of certain cross-sections surely introduces some bias in the comparisons and should be kept in mind.

3.2 Core model: INSTANT

3.2.1 Model of the core in INSTANT

Our core is symmetric, and so a $1/6^{th}$ core model has been built to speed up the calculations. This core model is used for all calculation schemes, no matter the DRAGON model that prepares the cross-sections. Thus there is no bias coming from different core models when we compare two methods of cross-section preparation.

The $1/6^{th}$ core is made up of 56 blocks of fuel and reflector. The main geometric dimensions are given in Table (3.1) and Figure (3.1) provides a drawing of the core.

Figure 3.1 – INSTANT $1/6^{th}$ core geometry - mixtures

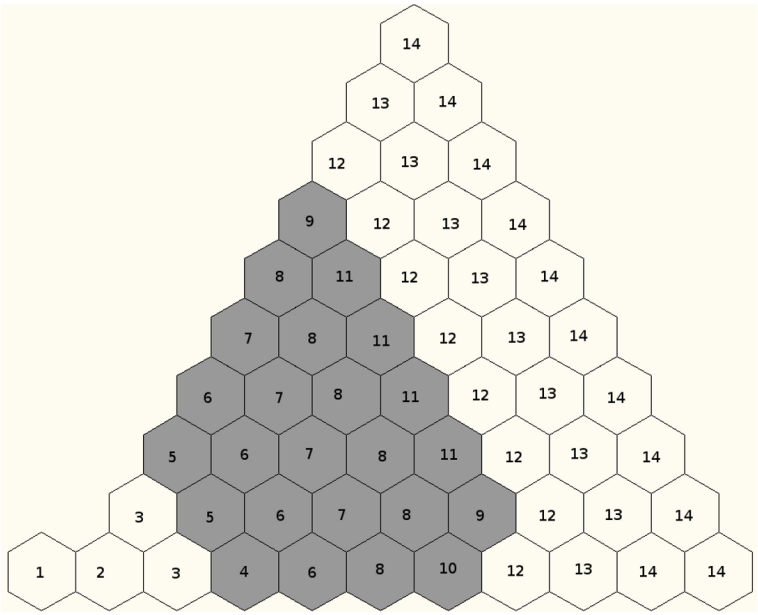


Table 3.1 – $1/6^{th}$ core - geometrical parameters in INSTANT

Parameter	Value
Block pitch (cm)	36,0
Block side (cm)	20,78461
N Maximum number of hexagons in a direction	21
T Total number of hexagons ($1/6^{th}$ core)	56

The INSTANT model allows the provision of different cross-sections in each row: 14

different mixtures are defined. Mixtures 1, 2, and 3 correspond to the inner reflector, whereas 12, 13, and 14 stand for the outer reflector. Those cross-sections will be generated once, and kept for every calculation in order to better isolate the impact of the different sets of fuel cross-sections (mixtures 4 to 11). Details about the generation of cross-sections for the reflector will be provided later.

The code uses the hybrid finite element method (FEM) for the spatial discretization and the P_N method for the angular discretization [24]. The interior and interface shape functions used for the FEM are polynomials. The solver is governed by 4 parameters: the interior polynomial expansion order p_{int}, the interface expansion order p_{surf}, the P_N order p_n and the source order expansion p_{sourc}, which controls the spatial expansion for the source calculation. Those parameters must respect the following rules in 2D-hexagonal geometry [1]:

$$\begin{aligned} &1 \leqslant p_n \leqslant 33 \text{ and } p_n \text{ odd number} \\ &p_{int} \geqslant p_{surf} + 3 \\ &0 \leqslant p_{sourc} \leqslant p_{int} \end{aligned} \tag{3.1}$$

The script to pass cross-sections from DRAGON to INSTANT permits to use only the isotropic component of the scattering source (scattering order 0), or to add a linear anisotropic component (scattering order 1). In this work, the linear anisotropic part is typically used, but some calculations have been performed to determine the importance of this order. The scattering cross-sections have been taken directly from the DRAGON output (SCAT00 and SCAT01). This way of handling the scattering is called the regular method in the following pages.

The script also permits the option of performing a calculation with the isotropic part of the scattering source but with "tricked" cross-sections, so that one may reproduce a sort of diffusion calculation. As explained in Section 2.12.3, this way of treating cross-sections is not completely equivalent to a diffusion calculation, but it provides indication on what it would give. An example is also given with this feature to assess by comparison with the regular transport calculation whether the core has a diffusive behaviour or not.

3.2.2 Convergence study of the core model

A convergence study has been performed to determine an acceptable set of parameters in the solver. For this purpose, cross-sections were prepared by the first DRAGON model (single block path). The K_{eff} has been taken as a convergence criterion. The effects of increasing p_{surf} order, p_{int} order, and p_{sourc} order were analysed. The study has been done for P_1 and P_3 calculations. It will then be shown how well it is converged in angles (P_N parameter).

P_1 convergence study, scattering order 1, regular method

The behaviour of p_{sourc} was first studied, setting $p_{surf} = 2$ and $p_{int} = 6$. Results are given in table 3.2(a). It is concluded that the convergence is ensured with $p_{sourc} = 5$. This value is kept for the next study on p_{surf} and p_{int}, the results of which are collected in table 3.2(b).

Table 3.2 – Convergence study for the P_1 core model

(a) Study on p_{sourc}

p_{sourc}	Core K_{eff}
0	1,20008
1	1,25369
2	1,25068
3	1,25054
4	1,25049
5	1,25046
6	1,25046

(b) Study on p_{surf} and p_{int}

p_{int}	$p_{surf} = 2$ K_{eff}	$p_{surf} = 3$ K_{eff}	$p_{surf} = 4$ K_{eff}
5	1,25046		
6	1,25046	1,25046	
7	1,25047	1,25047	1,25047

This shows that these settings ensure convergence, thus $p_{surf} = 2$ and $p_{int} = 6$ were used because the convergence is easier to achieve with only 44 iterations.

P_3 convergence study, scattering order 1, regular method

The same study has been repeated with a P_3 calculation using exactly the same material information.

Table 3.3(a) gives the results of the convergence study on p_{sourc} with $p_{surf} = 2$ and $p_{int} = 6$. The convergence is ensured with $p_{sourc} = 5$ and this value is kept for the next study on p_{surf} and p_{int} (table 3.3(b)). We finally keep $p_{surf} = 2$ and $p_{int} = 6$ because the convergence is easier to achieve with those values (95 iterations, see table 3.3(b)).

In both cases, the default refinement of the mesh proved to be sufficient to reach convergence. Further discretization does not affect the solution.

P_5 calculation, scattering order 1, regular method

Finally, a P_5 calculation was performed with those settings. This produces the same K_{eff} as with a P_3 calculation with less than 1 pcm of difference. Those results are given in appendix E. Therefore, it is concluded that a P_3 calculation gives a completely converged model and it is thus kept for the remainder of the study. Some examples in P_1 will be provided in appendix E to show the variability of the core calculation according to this parameter.

In summary, the following settings were used in INSTANT:

Table 3.3 – Convergence study for the P_3 core model

(a) Study on p_{sourc}

p_{sourc}	Core K_{eff}
0	1,20132
1	1,25349
2	1,25046
3	1,25030
4	1,25026
5	1,25023
6	1,25023

(b) Study on p_{surf} and p_{int}

p_{int}	$p_{surf} = 2$ Core K_{eff}	$p_{surf} = 3$ Core K_{eff}	$p_{surf} = 4$ Core K_{eff}
5	1,25024		
6	1,25023 (95 it.)	1,25023 (partial convergence)	
7	1,25023	1,25023	1,25023 (1439 it.)

– P_3 calculation
– $p_{source} = 5$; $p_{surf} = 2$; $p_{int} = 6$;
– default mesh refinement.

3.2.3 Scattering order and diffusion imitation

A calculation was also performed using the same cross-sections but with a scattering order reduced to 0 (isotropic scattering), in 26 groups. The following is obtained: $K_{\text{eff P3 scat0}} = 1,25258$, compared to $K_{\text{eff P3 scat1}} = 1,25023$. Both have been done in P_3. With 188 pcm of difference, it is concluded that the importance of the linearly anisotropic part of the scattering is small but not totally negligible.

Another trial was performed in P_1 by "tricking" cross-sections with the transport correction so that it approaches the results of a diffusion code. This resulted in $K_{\text{eff P1 scat0 diff}} = 1,25056$. This is very close to $K_{\text{eff P3 scat1}} = 1,25023$, so it is concluded the core is rather diffusive. The graphite moderator is known to have a rather isotropic behaviour [8], so this result is expected.

3.3 Single hexagonal fuel cell: DRAGON, SERPENT and MCNP

3.3.1 Effect of the boundary condition, homogeneous cell

Hexagonal cell

A single hexagonal cell containing a homogeneous fuel pellet was first simulated with DRAGON and SERPENT. The difference of modelling is related to the boundary conditions.

This study was performed because DRAGON is only able to set a white boundary condition, whereas MCNP can use either a white boundary condition or a true specular reflective

condition. It is admitted that a reflective boundary condition is more representative of the reality than a white condition. This study is therefore used to assess the importance of the type of boundary condition on the simulation, even if DRAGON is not offering any choice for hexagonal geometries.

The Monte-Carlo code SERPENT was used to compare to DRAGON because it has a double-heterogeneity module allowing one to better isolate the effect of the boundary condition. This would not have been the case with MCNP where the TRISO treatment may be sometimes questionable.

The SERPENT simulation was performed by Nicolas Martin (École Polytechnique de Montréal) with a library based on the JEFF-3.1.1 evaluation. The following result was obtained:

$$K_{\text{eff SERPENT hexcell}} = 0,97858 \pm 22 \text{ pcm}. \tag{3.2}$$

This serves as the reference. The simulation was performed with a reflective condition at the boundary.

In DRAGON, one is only able to set a white reflective boundary condition for hexagonal geometries. The calculation is done with the MOC technique. The basic calculation uses 2 iterations in the self-shielding, with correlation between the fissile isotopes and a transport correction. Table 3.4 details the results obtained for different parameters.

Table 3.4 – Hexagonal cell: different set of parameters

Parameter			Keff
TISO		USS / geometry	
nangl	dens		
12	20	PASS 3	0,97585
12	20	NOTR	0,97585
12	20	NOCO	0,97586
12	20	no sect, 4 radii	0,97588
12	30	no sect, 7 radii	0,97551
12	40	no sect, 7 radii in fuel, 2 radii in graphite	0.97557

We remark that the results are located between 0,97551 and 0,97588. It means about 300 pcm of difference compared to the SERPENT reference, and indicates that the different boundary condition seems to have a big effect. However, it is also admitted that the larger the lattice is, the smaller is the difference between the two types of boundary conditions. Therefore, a larger lattice was created in DRAGON with a $1/12^{\text{th}}$ symmetry, and with a central cell and 8 rings of fuel cell. There are 25 cells in the assembly, or 217 if we had taken

the complete geometry. With this lattice, and TISO 12 20,0, we obtain:

$$K_{\text{eff DRAGON hexcell}} = 0,97854. \tag{3.3}$$

This is 4 pcm less than SERPENT.

In conclusion, this study shows a very good agreement between the white and the reflective conditions for a large domain, whereas a white condition on a single hexagonal cell leads to an error of 300 pcm down compared to SERPENT. It therefore demonstrates that with large lattices, the two conditions have quite a similar effect. As our block model will have 11 rings of cells, a white boundary condition may be sufficient to perform the calculation. A white boundary condition is set in MCNP, too, to remain consistent in comparisons.

Square pin cell

A second study of the boundary conditions was performed in DRAGON on square cells in order to assess this effect using the same code. The hexagonal cell was changed into a square, so that a cyclic tracking can be performed in DRAGON. The square had a side of 1,7495 cm, so as to conserve the same area between the two DRAGON models.

With tracking parameters 16 60,0 and a reflective boundary condition in DRAGON, $K_{\text{eff DRAGON}} = 0,97978$ is obtained and the maximal error on the volumes is equal to 0,8 %. Table 3.5 details some other choices for the parameters. It is shown that the model is converged.

Table 3.5 – Square cell: different set of parameters

Parameter			Keff
TISO		Nb. of radii in the fuel pellet	
nangl	dens		
12	20	4 radii	0,98024
12	30	4 radii	0,97982
12	40	7 radii	0,97995
12	50	6 radii	0,97977
12	60	6 radii	0,97978

The SERPENT calculation for this case gives $K_{\text{eff SERPENT}} = 0,97916 \pm 44$ pcm. Another Monte-Carlo calculation in DRAGON using the new stochastic module MC of DRAGON has also been performed by Nicolas Martin. He was thus able to use the same library as in the deterministic calculation, and the same self-shielding parameters. We get:

- $K_{\text{eff}} = 0,97942 \pm 55$ pcm with a correlated self-shielding
- $K_{\text{eff}} = 0,97970 \pm 55$ pcm without correlation in the self-shielding.

We observe that the K_{eff} are all located in the same range of $\pm$ 30 pcm, that is to say a statistically negligible difference.

On the contrary, a white condition would introduce a bias of about 300 pcm. In DRAGON, a tracking with 16 60 and a white boundary condition gives $K_{eff} = 0{,}97615$.

We conclude that in both geometries, *for a single cell*, the type of boundary condition is important for the K_{eff} value. A white condition will introduce a bias of 300 pcm on small domains compared to a reflective condition. However, as shown in the previous study, the increase of the domain leads to the convergence of the results using the two different boundary conditions. Even if the cyclic tracking has not yet been coded into DRAGON for hexagonal cells, it can be assumed that the result would be similar in this case.

3.3.2 Effect of the surrounding graphite on the energy self-shielding

Finally a study was performed to evaluate the effect of the graphite number density in the region surrounding the pellet on the energy self-shielding calculation. This is of interest because self-shielding calculation will not be performed on the supercell models. For these, the self-shielding calculation will be performed on a block without its environment in a pre-calculation before passing homogenized cross-sections to the supercell calculation in DRAGON. Therefore, it is of interest to know if this can be a significant source of error compared to MCNP where the self-shielding is implicitly accounted for the entire domain. As deep burn fuel is abundant in resonant absorbers, the energy self-shielding may depend on the fuel block environment.

The difficult part of this study is that the effect of having more graphite outside the fuel pellet has to be studied only on the energy self-shielding calculation. It would be difficult to study cross-sections in 295 groups, therefore a condensation is needed, but this operation requires a flux. However, an increase of the graphite density will produce a different flux calculation, which would prevent one to isolate the self-shielding effect. This is why something more special has to be performed.

Here is the principle of the study. A first complete calculation is performed on a single cell. The self-shielding calculation is performed, then the flux calculation, then homogenized and condensed cross-sections are produced. This flux calculation is saved. The homogenized and condensed cross-sections are held as a reference.

Then the same geometry is taken but the graphite number density is increased by a factor of 100 outside the fuel compact. The self-shielding is recalculated, *but not the flux calculation.* Instead, the flux previously saved is called back and used to homogenize the cross-sections. Thus, those cross-sections are being differently self-shielded, but they are homogenised and condensed by the same weighting function (i.e. the flux). One is then able to compare them

in 6 groups. The 6-group structure is given in appendix F. If the self-shielding has changed, then it should appear in the cross-sections.

Table 3.6 gives the cross-sections obtained with a normal regular graphite density in the region surrounding the fuel pellet whereas table 3.7 gives the same cross-sections when the graphite number density has been multiplied by 100. It is shown that there are almost no differences. The maximum difference is of 0,7 % of the regular cross-section on the absorption cross-section in group 2.

Table 3.6 – Single cell - Homogenized cross-sections with a normal graphite density around the fuel pellet

Homogenized cross-sections – Normal density					
Group	Total XS	Absorption XS	$\nu\Sigma_{fission}$	Scattering XS within group	Scattering XS out of group
1	2,4540E-01	2,0650E-04	4,8208E-04	2,2969E-01	1,5501E-02
2	3,9499E-01	1,8231E-03	2,1290E-03	3,8701E-01	6,1570E-03
3	4,0725E-01	8,9779E-03	1,1225E-02	3,6992E-01	2,8350E-02
4	4,5111E-01	4,8804E-02	5,4703E-03	3,6158E-01	4,0726E-02
5	4,7317E-01	7,3240E-02	1,1485E-01	3,8322E-01	1,6712E-02
6	4,6411E-01	5,5374E-02	9,7179E-02	4,0043E-01	8,3055E-03

Table 3.7 – Single cell - Homogenized cross-sections with an increased graphite density by 100 around the fuel pellet

Homogenized cross-sections – 100 × Normal density					
Group	Total XS	Absorption XS	$\nu\Sigma_{fission}$	Scattering XS within group	Scattering XS out of group
1	2,4540E-01	2,0650E-04	4,8208E-04	2,2969E-01	1,5501E-02
2	**3,9501E-01**	**1,8362E-03**	**2,1359E-03**	**3,8702E-01**	6,1570E-03
3	4,0725E-01	8,9779E-03	1,1225E-02	3,6992E-01	2,8350E-02
4	4,5111E-01	4,8804E-02	5,4703E-03	3,6158E-01	4,0726E-02
5	4,7317E-01	7,3240E-02	1,1485E-01	3,8322E-01	1,6712E-02
6	4,6411E-01	5,5374E-02	9,7179E-02	4,0043E-01	8,3055E-03

It is concluded that the self-shielding is not really affected by the surrounding graphite density. This result was quite unexpected, because the neutron slowing-down may have been modified by the graphite, but it seems that it is not the case. The energies where the self-shielding is taking place may be too high to see the influence of the graphite.

3.4 Single fuel block model: DRAGON

The first scheme of calculation begins with the modelling of a single block of fuel in DRAGON.

In MCNP, the calculation over the real block was performed and is held as a reference. The following result was obtained:

$$K_{\text{eff SB MCNP}} = 1,25969 \pm 0,00012. \tag{3.4}$$

3.4.1 Description of a fuel block and boundary approximation:

The real fuel block is comprised of 11 rows of hexagons, plus the central hexagonal cell. Figure (1.2) gives a representation of the geometry. Table 1.2(a) gives the dimensions considered.

Special attention must be paid to the boundary cells. Indeed, DRAGON does not have the capability to include a portion of a hexagonal lattice into a larger hexagon. Thus, it is necessary to terminate the lattice by a row of hexagons with an adjusted graphite density. This density has been determined so as to conserve the total number of graphite atoms in the whole core calculation (see appendix C). This solution has been also used by other authors confronted by the same problem (see [25, 26, 27, 28]) and it appears to be accurate.

In DRAGON, a homogenization of the whole cell is performed and the resulting microscopic cross-sections are used in INSTANT. The densities are corrected to fit with the volume of a block in INSTANT (see appendix D).

3.4.2 Spatial convergence study of the single-block model

DRAGON enables the user to set many different parameters. In particular, it is necessary to choose the level of discretization in order to ensure that the mesh is sufficient for this problem. Two tracking modules were tested: `EXCELT:` and `SYBILT:` [3].

`EXCELT:` tracking and MOC

A first possibility is to use the method of characteristics to solve the Boltzmann equation. This is done in DRAGON with the `EXCELT:` tracking module followed by the module MCCGT which converts the tracks for the MOC solver.

A convergence study over all the parameters together would require a huge amount of calculations. But the parameters seems to be partially decoupled, so that they can be studied independently. By setting relative high values for the parameters that are fixed while others

are varied, convergence criteria can be met. For the double-heterogeneity model, the Hébert model will be used for the whole study [29, 30].

We first try to find whether a discretisation of the fuel pellet is needed. The annular region of helium is not studied because helium is basically neutron transparent. The number of radii in table 3.8(a) corresponds to the number of radii inside the fuel region. 1 radius means that the fuel pellet is not radially discretized. For those calculations, relative high values were set for generating the tracking lines: 4 angles and 40 lines per cm. It was concluded from these results that there is no need to discretize the fuel pellet, but the division of the hexagonal fuel cell into 6 sectors counts for 10 pcm and will be kept in further calculations. The division of hexagons into sectors is called sectorization or azimuthal sectorization in the remainder of the thesis. It is abbreviated by "Azimuth. sect." in the tables.

Table 3.8 – Convergence study for the single block: geometrical mesh

(a) Fuel cell radial discretization

Parameters			K-eff
Cell type	Number of radii	Azimuth. sector.	
Fuel	4	yes	1,25919
Graphite	1	yes	
Coolant	0	no	
Fuel	3	yes	1,25918
Graphite	1	yes	
Coolant	0	no	
Fuel	2	yes	1,25915
Graphite	1	yes	
Coolant	0	no	
Fuel	1	yes	1,25911
Graphite	1	yes	
Coolant	0	no	
Fuel	1	no	1,25902
Graphite	1	yes	
Coolant	0	no	

(b) Graphite cell radial discretization

Parameters			K-eff
Cell type	Number of radii	Azimuth. sector.	
Fuel	1	yes	1,25913
Graphite	1	yes	
Coolant	1	yes	
Fuel	1	yes	1,25908
Graphite	0	no	
Coolant	0	no	
Fuel	1	yes	1,25908
Graphite	1	no	
Coolant	0	no	

A study was then performed to determine the influence of the discretization in the graphite and coolant regions. For this, the fuel pellets remain sectorized. In graphite cells, no radius means that we use a regular hexagonal cell without any pellet. The parameters TISO 4 40,0 were kept for the tracking. Results are shown in table 3.8(b).

Table 3.8(b) shows that neither the discretization nor the sectorization of the graphite regions are significantly changing the K_{eff}. It was concluded that discretization of the graphite

cells is unnecessary. But as these results were not complete at the beginning of the study, the next calculations were performed with sectorized graphite cells containing 1 radius.

A study over the tracking parameters was completed to determine the minimum that could be used with acceptable accuracy. Table 3.9(a) shows the density of lines that has to be used and table 3.9(b) considers the number of angles.

Table 3.9 – Convergence study for the single block: tracking parameters

(a) Density of lines

TISO		K-eff
Number of angles	Density (lines.cm^{-1})	
4	100	1,25911
4	50	1,25911
4	40	1,25911
4	30	1,25911
4	20	1,25913
4	10	1,25918

(b) Number of angles

TISO		K-eff
Number of angles	Density (lines.cm^{-1})	
12	30	1,25911
10	30	1,25928
8	30	1,25928
6	30	1,25827
4	30	1,25911

It can be concluded that TISO 4 10 creates enough integration lines to cover the whole geometry and produce an efficient integration. The parameters TISO 4 10 were used for all the proceeding calculations. This allows the tracking to place 6 segments in the fuel pellet, which appears to be enough.

The QUAB parameter defines the number of points used to calculate the integrals inside the TRISO particles. The study's results are summarized in table 3.10(a). It is concluded that a 5-point basis for the integration in the TRISO particles works well. A Gauss quadrature is always used because it ensures the best accuracy for the numerical integration.

Table 3.10 – Convergence study for the single block: integration points in `BIHET:` and self-shielding parameters

(a) Integration points in TRISOs

QUAB	K-eff
7	1,25919
6	1,25919
5	1,25918
4	1,25911
3	1,25874
2	1,25107

(b) Self-shielding parameters

Parameters for the self-shielding			K-eff
PT accuracy	Number of external iterations	Correlation	
3	2	yes	1,25918
3	3	yes	1,25918
2	2	yes	1,25918
3	2	no	1,25914

Finally, several parameters were evaluated in the self-shielding module USS, for example the number of external iterations, the accuracy of the probability tables and the correlation of the isotopes. The variation of K_{eff} was used as a figure of merit for the convergence. Table 3.10(b) gives the results.

This shows that having 3 and 2 external iterations gives a good representation of the self-shielding. It also shows that the correlation effect between the fissile isotopes is low. This does not automatically imply that all isotopes are independent, but that the coupling effect is small.

In conclusion, the model chosen for an `EXCELT:` tracking of the single block model is made of:

- Geometry: sectorization of the fuel pellet without any radial discretization, no discretization nor sectorization of the graphite and coolant cells;
- Tracking: TISO 4 10, QUAB 5;
- Self-shielding: 2 iterations (PASS 2), probability tables with an accuracy of 3, no correlation between fissile isotopes during the self-shielding.

With this model, the following K_{eff} is obtained:

$$K_{\text{eff SB MOC}} = 1,25911. \tag{3.5}$$

Compared to MCNP, it is -46 pcm lower. Thus, the agreement between the two codes is very good.

`SYBILT:` tracking and collision probabilities method

The other module available to do the calculation in hexagonal geometry is called `SYBILT:` and uses the collision probability method with interface currents coupling.

The interface current method is an alternative to the traditional collision probability (CP) method for considering several uncoupled regions instead of one large coupled region. The calculation is performed in each region separately. The regions are then recoupled together by the knowledge of the interface currents on each edge surrounding each cell as described in reference [6]. An iterative scheme shall converge to the solution satisfying both inner transport equations and interface coupling currents. Those currents obey a balance equation: the outgoing current of cell A into cell B is equal to the incoming current from cell B to cell A.

The module is very fast but less reliable than `EXCELT:` because it uses more approximations. In `SYBILT:`, parameters must be set for the geometrical mesh, the integration lines and the number of self-shielding iterations. The technique of interface current method was

used with a double P_1 expansion of the surface flux surrounding the hexagonal cells (option DP01 in DRAGON).

A convergence study was first performed over the geometric specifications of the mesh: radial discretisation and sectorization of cells. For this study, 3 iterations were used for the self-shielding. The integration line parameters were set to: 8 angles, 10 segments per line (QUA2 8 10) and 5 segments in TRISO particles (QUAB 5). Table 3.11(a) contains the results for the discretization of the fuel cells and table 3.11(b) contains the same information for the graphite and coolant cells. Table 3.11(b) gives also the results of different trials to determine whether the graphite and coolant cells had to be sectorized or not.

Table 3.11 – Convergence study for single block: geometrical mesh

(a) Fuel cell radial discretization

Parameters			K-eff
Cell type	Number of radii	Azimuth. sector.	
Fuel	9	yes	
Graphite	1	yes	1,26052
Coolant	1	yes	
Fuel	7	yes	
Graphite	6	yes	1,26052
Coolant	1	yes	
Fuel	4	yes	
Graphite	6	yes	1,26050
Coolant	1	yes	
Fuel	3	yes	
Graphite	6	yes	1,26048
Coolant	1	yes	
Fuel	2	yes	
Graphite	6	yes	1,26045
Coolant	1	yes	
Fuel	1	yes	
Graphite	6	yes	1,26038
Coolant	1	yes	

(b) Graphite cell radial discretization, coolant and graphite cells sectorization

Parameters			K-eff
Cell type	Number of radii	Azimuth. sector.	
Fuel	4	yes	
Graphite	6	yes	1,26050
Coolant	1	yes	
Fuel	4	yes	
Graphite	4	yes	1,26049
Coolant	1	yes	
Fuel	4	yes	
Graphite	2	yes	1,26049
Coolant	1	yes	
Fuel	4	yes	
Graphite	1	yes	1,26049
Coolant	1	yes	
Fuel	4	yes	
Graphite	0	none	1,26047
Coolant	1	yes	
Fuel	4	yes	
Graphite	1	yes	1,26048
Coolant	1	none	
Fuel	4	yes	
Graphite	0	none	1,26046
Coolant	1	none	

It was concluded that the geometry is sufficiently discretized when there are 3 radii in the fuel pellet, no sectorization and no radial discretisation in the coolant cells and graphite cells. However, these studies had not been completed at the outset so the following parameters

were used for further convergence studies: 3 radii and sectorization in the fuel pellet, no sectorization in the other cells, and 1 radius with sectorization in the graphite cells.

The next study was used to determine the parameters for the integration lines that will be adapted to the geometry. Those parameters are called QUA2 and QUAB in DRAGON. QUA2 refers to the macroscopic geometry, whereas QUAB refers to the double-heterogeneity model. It is recalled here that the choice of each parameter is performed independently from each other. 3 iterations were kept for the self-shielding. The results are given in table 3.12. It shows that QUA2 4 8 is a good combination.

Table 3.12 – Convergence study for single block: integration lines

Parameters			K-eff
QUA2		QUAB	
Nb. angles	Nb. segments		
8	10	5	1,26047
6	10	5	1,26047
4	10	5	1,26046
3	10	5	1,26052
2	10	5	1,26020
8	10	5	1,26047
8	8	5	1,26046
8	6	5	1,26044
8	4	5	1,26042
8	3	5	1,26040
8	10	8	1,26047
8	10	7	1,26047
8	10	5	1,26047
8	10	3	1,26003

A study over the self-shielding iterations had also been performed. The parameters were set to QUA2 8 10 and QUAB 5, in order to be fully converged at the geometrical level. Table 3.13(a) shows the results for different options. This shows that the self-shielding iterations are totally converged after 2 iterations. The effect of the correlation between the fissile isotopes was also studied. Until now, it has been assumed that there was a mutual self-shielding between all of the resonant isotopes that had to be taken into account. On the contrary, assuming that all isotopes can be self-shielded independently may improve the calculation time, but it must be verified that it does not affect K_{eff} (see table 3.13(b)).

Table 3.13 – Convergence study for the single block: self-shielding

(a) Number of self-shielding iterations

Parameters	K-eff
Nb. of self-shielding iterations	
5	1,26047
4	1,26047
3	1,26047
2	1,26047
1	1,26046

(b) Self-shielding correlation

Parameter	K-eff
Correlation	
yes	1,26043
no	1,26039

In conclusion, the following model was conserved for all single block calculations with `SYBILT:`:

- Geometrical discretization:
  - fuel cells (F): 3 radii inside the pellet (0.0 0.207 0.415 0.6225) and sectorization;
  - coolant cells (K, C): no radial discretization, no sectorization;
  - graphite cells (G, J): no radial discretization, no sectorization;
- Tracking:
  - 4 basis points for the angular integration of the blocks
  - 8 basis points for the spatial integration of the blocks
  - 5 basis points for the numerical integration of the collision probabilities in the micro-volumes using the Gauss-Jacobi formula. Hébert double-heterogeneity model is used (QUA2 4 8 QUAB 5 HEBE).
- Self-shielding: 2 external flux iterations for the self-shielding calculation, and handle all fissile isotopes as non-correlated;
- Iterative techniques were used to compute the self-shielding and the collision probabilities.

With those parameters, the following multiplication factor is obtained for the single block:

$$K_{\text{eff SB SYB}} = 1,26039. \tag{3.6}$$

It is +56 pcm higher than the MCNP result. The agreement is comparable to the method of characteristics (`EXCELT:`).

Conclusion: Summary for the single block calculations:

Table (3.14) summarizes the results obtained on the single block model with the parameters selected for the two trackings.

Table 3.14 – Convergence study for the single block: summary:

Code	Tracking module	Solution method	K-eff	Diff. to MCNP
MCNP	\	Monte-Carlo	1,25969 ± 0,00012	\
DRAGON	`SYBILT:`	Pij	1,26039	+56 pcm
DRAGON	`EXCELT:`	MOC	1,25911	-46 pcm

The single block model is used to produce homogenized cross-sections over the whole block, condensed to different numbers of energy groups. It is also used to produce cross-sections for other DRAGON calculations which avoid some details: for example in the supercell calculations, for which cross-sections are partially homogenized, or for the reflector calculation. In those cases, the `EXCELT:` tracking is used because it is more accurate than the `SYBILT:` one.

3.5 Reflector model: DRAGON

3.5.1 Description of the reflector model

The reflector graphite properties for INSTANT have been calculated with a simplified 2D cylindrical model of the whole core. Fuel homogenized cross-sections were first produced in 295 groups using the single block model, and then a cylindrical core geometry was used whose dimensions were determined such that the total area of the real geometry for each annular region is conserved. Table 3.15 shows the dimensions used. The number densities are the one corresponding to the INSTANT model, which ensures consistency. A void boundary condition is set.

The INSTANT calculation can be performed in P_3 or in P_1 approximation. In P_1, a sort of diffusion approximation in multigroup can be performed, as explained in section 2.12.3, page 48. However, in order to do this, the transport cross-sections in DRAGON are required, which can only be produced with a white boundary conditions. Therefore, an extra ring was added 5,165 cm thick filled with ^{10}B at the periphery with a white boundary condition. This simulates the void condition because the density of ^{10}B is very high, so it can be assumed that the neutrons will be absorbed. This is probably not strictly true for the fast neutrons, and it is a source of error. The other solution is to work with a regular P_3 method in INSTANT.

Table 3.15 – Determination of the corresponding radii for the reflector regions

Region	Type of material	Nb. hex.	Cumulated area (cm^2)	Adjusted radius (cm)
1	Reflector	1	1128,61	18,954
2	Reflector	7	7900,29	50,147
3	Reflector	19	21443,64	82,618
4	Fuel	169	190735,59	246,400
5	Reflector	217	244909,014	279,208
6	Reflector	271	305854,11	312,020
7	Reflector	331	373570,89	344,835

For this method, one may use a void condition at the periphery, and therefore the extra ring of ^{10}B would not be needed. Both have been done, which allowed to perform both types of calculations in INSTANT.

The reflector properties will be kept the same for the whole study, so as not to introduce a bias while recalculating them with different methods or different cross-sections for the fuel rings. This study focuses only on the preparation of the fuel homogenized cross-sections for INSTANT. Later, it would be naturally possible to evaluate other methods of modelling the reflector.

For the same reason, one set of homogenized fuel cross-sections must be chosen to perform all reflector calculations. It was decided to use the `EXCELT:` tracking to produce the fuel cross-sections for the reflector calculation because this module is considered as more accurate.

The reflector calculation was not validated using MCNP. Nevertheless, the `SNT:` solver has been tested in other projects at Montréal and has proven to be reliable. The K_{eff} of this model should also be relatively close to the K_{eff} of the real $1/12^{th}$ core.

3.5.2 Convergence study of the reflector model

A convergence study was necessary to choose the parameters in DRAGON for the reflector model, especially the spatial discretization. Two solvers are available in Cartesian geometry: the `SNT:` solver and `SYBILT:`. The criterion to choose between them was convergence of K_{eff}. The study was performed with the extra ring of ^{10}B so that a white boundary condition could be used and transport cross-sections produced. The study was not duplicated for both the real and simulated void conditions.

Convergence study for the `SNT:` tracking

Table 3.16(a) gives the results of different discretizations of the annular regions. G stands for the graphite reflector, F for the fuel annular region and ^{10}B for the external layer of boron. In DRAGON, the option SPLITR with a negative number enables a mesh splitting along the radial direction into zones of equal volumes. Other DRAGON parameters were set to: SN14; LIVO 10 5; MAXI 100; EPSI 1E-05; QUAD10.

Table 3.16 – Convergence study for the reflector: geometry discretization and angular discretization

(a) Spatial discretization

SPLITR: Number of volumes in region:								K_{eff}
1	2	3	4	5	6	7	8	
G	G	G	F	G	G	G	^{10}B	
-15	-15	-25	-50	-25	-15	-15	-2	1,25208
-10	-10	-20	-50	-20	-10	-10	-5	1,25207
-20	-7	-8	-30	-8	-9	-10	-10	1,25204
-10	-10	-20	-20	-20	-10	-10	-5	1,25206
-5	-5	-10	-20	-10	-5	-5	-5	1,25205
-20	-7	-8	-15	-8	-9	-10	-10	1,25201

(b) S_N order

S_N	K_{eff}
22	1,25206
18	1,25206
14	1,25206
10	1,25206
6	1,25205
4	1,25205

From this, one sees that the results are not strongly influenced by the spatial discretization. A good compromise between the precision and the calculation time seems to be: SPLITR -10 -10 -20 -20 -20 -10 -10 -5, each number corresponding to the number of divisions in each region. This leads to a K_{eff} of 1,25206 which is close to the K_{eff} obtained with the highest discretization (1,25208). This was kept for further calculations.

The number of angles taken to do the calculation (S_N) was then studied. Results are given in table 3.16(b). The Eigenvalues are particularly stable even with a low number of angles. This comes from the cylindrical symmetry. An order of 14 was kept for the next calculations, but 10 or even 6 would likely have been sufficient. However, S_{14} enabled the use of previous calculations again for this study. For the final model, S_{10} was used. Calculations with two different angular quadratures types were then performed, whose results are given in table 3.17.

The two quadratures gives almost the same results. This parameter might have more influence in 3-D geometries. The quadrature number 10 was kept: a product of the Gauss-Legendre and the Gauss-Chebyshev quadrature.

The effect of the Livolant acceleration method (table 3.18) was also examined.

Except for the trial with 20 accelerated iterations, all the Eigenvalues were the same. It

Table 3.17 – Convergence study for the reflector: angular quadrature type

Quadrature	K-eff
4: Legendre-Chebyshev quadrature	1,25209
10: product of Gauss-Legendre and Gauss-Chebyshev quadrature	1,25206

Table 3.18 – Convergence study for the reflector: Livolant acceleration

LIVO: Livolant acceleration		K_{eff}	Nb. of outer iterations	Calculation time (s)
Nb. of free iterations	Nb. of accelerated iterations			
30	5	1,25206	79	1434
20	5	1,25206	97	1046
10	5	1,25206	78	1347
5	5	1,25206	44	674
10	25	1,25206	35	727
10	20	1,25211	59	757
10	15	1,25206	68	800
10	10	1,25206	86	1120

was thus concluded that the case with 20 free-iterations may have had numerical issues and may not be relevant. 10 accelerated iterations and 5 free iterations were kept for the next calculations, but it seemed that the best choice for the final model would be to take 5 free iterations and 25 accelerated iterations. A trial with this setting reveals good results (see equation 3.7 below).

Finally, the effect of allowing different maximum numbers of thermal iterations in the flux solution (FLU module) was studied. The results are given in table 3.19. It shows no variation on the K_{eff}, but the number of thermal iterations has an effect on the duration of the calculation. 5 thermal iterations will be used in order to keep calculations as fast as possible.

The following settings were kept for all calculations on this geometry:

– SPLITR -10 -10 -20 -20 -20 -10 -10 -5
– SN 10 (about 100 angles)
– QUAD 10 (quadrature: product of Gauss-Legendre and Gauss-Chebyshev)
– LIVO 5 25
– THER 5 1.10^{-5}

We obtain:

$$K_{\text{eff Refl SN}} = 1,25206. \tag{3.7}$$

Table 3.19 – Convergence study for the reflector: number of thermal iterations

THER: maximum number of thermal iterations	K_{eff}	Nb. of outer iterations	Duration (s)
5	1,25206	72	828
10	1,25206	78	1347
15	1,25206	92	1719
20	1,25206	88	1538
30	1,25206	75	1036
40	1,25206	78	1520

This is about 194 pcm higher than the K_{eff} obtained in MCNP for the $1/12^{th}$ core (see section 4.1 page 89), which suggests that the reflector model is adequate. The external convergence is reached after 61 iterations and the calculation time is equal to 343 s in the FLU module.

Convergence study for the `SYBILT:` tracking:

The same type of study was performed with a `SYBILT:` tracking and the collision probability method. However, it was necessary to put a smaller concentration of boron at the periphery to simulate the void boundary condition. Otherwise, DRAGON was unable to perform the calculation due to excessively small collision probabilities in the boron region. The simulation ran to completion with a concentration of $8,04.10^{-1}.10^{24}$atoms.cm^{-3}. A higher concentration lead to convergence difficulties. The higher the concentration of boron is, the better the simulation of the void condition may be, so it was decided not to decrease the concentration used with the `SNT:` tracking. This boron concentration is anyway completely arbitrary and virtual, so the model whose K_{eff} convergence is the best was kept. A linear anisotropic model for the interface currents (DP01) was used, because this feature may increase the accuracy of the solution.

The spatial discretization was first studied. Results of several settings are provided in table 3.20.

It was observed that the convergence is quite poor with this solver, even when the geometry is highly discretized. `SYBILT:` may have problems to converge with a large number of regions. The discretization SPLITR -10 -10 -20 -25 -20 -10 -10 -5 was kept for the following calculations, but it is known that it is not well converged.

The number of integration points needed was then examined (table 3.21(a)).

Seeing the results, five integration points were kept: QUA1 5. This ensured a good convergence and minimized the calculation time. The effect of different maxima for the thermal iterations was then examined (table 3.21(b)). As one can see, there are not significant differ-

Table 3.20 – Convergence study for the reflector: spatial discretization

SPLITR: Number of volumes in region:								K-eff
1	2	3	4	5	6	7	8	
G	G	G	F	G	G	G	^{10}B	
-15	-20	-35	-50	-35	-30	-20	-5	1,24737
-15	-20	-30	-45	-30	-20	-20	-5	1,24718
-10	-10	-25	-40	-25	-10	-10	-5	1,24470
-10	-10	-20	-30	-20	-10	-10	-5	1,24511
-10	-10	-20	-20	-20	-10	-10	-5	1,24533
-20	-7	-8	-15	-8	-9	-10	-10	1,24784
-5	-5	-10	-15	-10	-5	-5	-5	1,23869
-2	-5	-10	-10	-10	-5	-2	-1	1,23237
0	0	0	0	0	0	0	0	1,11115

Table 3.21 – Convergence study for the reflector: integration parameters and number of thermal iterations

(a) Number of integration points

QUA1: number of integration points	K-eff
20	1,24522
15	1,24520
10	1,24521
8	1,24523
5	1,24523
3	1,24522

(b) Number of thermal iterations

THER: maximum number of thermal iterations	K-eff	Nb. of outer iterations	Duration (s)
5	1,24522	34	27
10	1,24523	44	36
20	1,24522	36	31
30	1,24521	32	29
40	1,24523	35	31
50	1,24523	35	31

ences, because usually the source convergence is reached after a small number of iterations. Therefore 5 thermal iterations were kept to minimize the calculation duration.

The final parameters chosen for a `SYBILT:` calculation on the reflector model were:

– SPLITR -10 -10 -20 -25 -20 -10 -10 -5
– QUA1 5
– DP01
– THER 5 EPSI 1.10^{-5}

These gives:

$$K_{\text{eff Refl SYB}} = 1,24522. \tag{3.8}$$

The calculation requires only 3 outer iterations and 27 s in the flux module. It confirms the speed of this solver. However, the solution is quite unstable regarding the spatial discretization: the Eigenvalue was not converged while discretizing the geometry. Therefore, the spatial convergence cannot be really insured. The Eigenvalue was also farther from the Eigenvalue calculated by MCNP on the $1/12^{\text{th}}$ core model: a difference of -352 pcm was observed between this and the MCNP model (see section 4.1 page 89).

Decision for the final model of reflector

Considering the results, it was decided to keep the `SNT::` tracking for all of the calculations requiring reflector properties. This solver may provide a better representation of the void boundary condition by increasing by 100 the boron concentration if this feature is needed, and its convergence is much more reliable than with `SYBILT:`. The cylindrical geometry may be particularly suitable for the SN method because the mesh proved to be stable with different spatial discretizations.

In conclusion, when the reflector calculation was performed with SN, an extra-ring of boron and a reflective condition at the periphery, it gave:

$$K_{\text{eff Refl void}} = 1,25206. \tag{3.9}$$

The same model without boron and with a void condition at the periphery gives:

$$K_{\text{eff Refl boron}} = 1,24656. \tag{3.10}$$

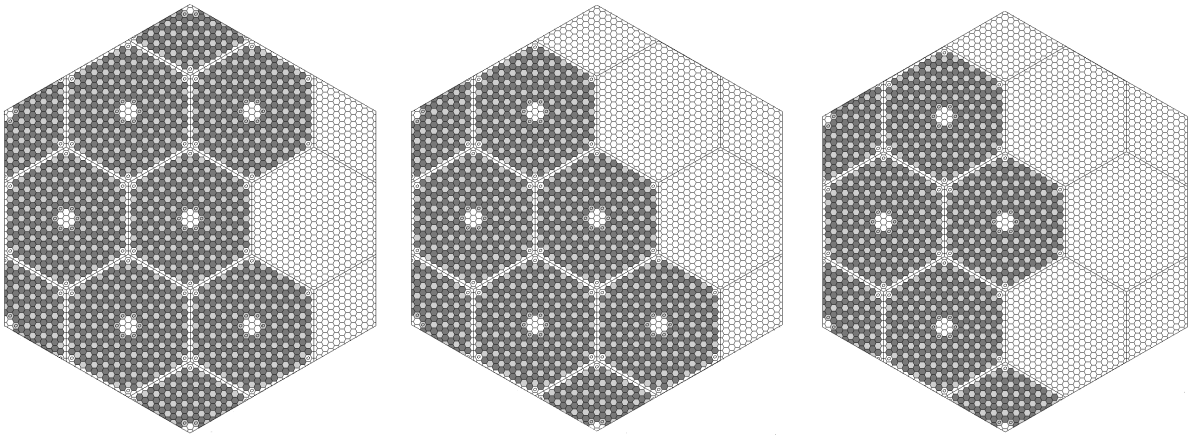
Both K_{eff} are quite close to the Eigenvalue of the $1/12^{\text{th}}$ core calculated by MCNP (+194 pcm and -247 pcm respectively), which suggests that this simplified model may be suitable to provide the reflector cross-sections.

3.6 Supercell model: DRAGON

3.6.1 Description of the supercell models

The supercell model attempts to find an intermediate path between the single block model and a full detailed $1/12^{\text{th}}$ core model. It consists of adding some external rings to the single block model so that the central block of interest is influenced by its neighbours. Therefore, the cross-sections recovered from this path should be weighted by a somewhat more realistic spectrum, but with a shorter calculation time than in the case of directly modelling a $1/12^{\text{th}}$ core with all details in DRAGON.

The supercells have 33 rings of cells, which corresponds to one row of neighbouring blocks next to the fuel block of interest. In total, the model contains 3367 hexagonal cells. Three different types of supercells were defined to take into account the various numbers of reflector blocks which each fuel boundary block is facing in the core model: either 1, 2 or 3 reflector blocks. Figure 3.2 shows the geometries considered. The block of interest is always the central fuel block, from which the homogenized cross-sections are recovered.



(a) 1 neighbouring reflector block (b) 2 neighbouring reflector blocks (c) 3 neighbouring reflector blocks

Figure 3.2 – Geometrical models for supercells

A full detailed calculation of this domain with compacts containing double-heterogeneity would require a large amount of memory and time. In addition, the version of DRAGON in use in this work uses an addressing system in 32 bits. This implies that the different objects created to store the data cannot exceed a size of about 2,1 Gb. This limitation forces the declaration of homogeneous cells for which homogenized cross-sections have to be prepared using the single block model. It also prevents the use of the method of collision probability with the `EXCELT:` tracking because the building of the Pij matrices requires more addresses than are available.

As a priori the interface current method in SYBILT: is not considered reliable for such a large domain, it was decided to keep the possibility to do this kind of calculation with the EXCELT: tracking and the method of characteristics, because the MOC solver is able to use iterative techniques instead of building full matrices.

The EXCELT: tracking in hexagonal 2D and 3D geometries is an implementation of the Ph. D. dissertation by Mohamed Ouisloumen (see reference [31]). At the origin, this tracking was used to compute collision probabilities and was not coupled with any double-heterogeneity model. The possibility of using the same tracking with the MOC was implemented in DRAGON Version4, together with full interoperability of two double-heterogeneity models. [2]

In order to reduce the calculation time, a single block calculation is performed first and the cross-sections are homogenized separately, depending on their location. Supercells are then constructed with cross-sections for the fuel pellet, the graphite surrounding the fuel pellet, the small coolant cells, the large coolant cells, the plain graphite cells and the graphite cells of the boundary whose densities are adjusted so that the same number of graphite atoms between the DRAGON models and reality (MCNP) is conserved. Keeping the fuel pellet inside its surrounding graphite instead of homogenizing the whole fuel cell may increase the accuracy, and it also makes the spatial convergence of the source easier.

Ideally, one would not condense the group structure after the single block calculation, so that the supercell calculation is performed with 295 groups. This is the usual way to do lattice calculations. However, keeping 295 groups for the supercell calculation is very time consuming, especially if a discretized geometry is used with the EXCELT: tracking module and the method of characteristics. A way to manage this problem consists of condensing the cross-sections to 26 groups after the single block calculation and performing the supercell calculation with only 26 groups. The consequences of this are evaluated in chapter 4. However, it may not be adequate if depletion calculations are needed.

After the supercell calculation, the block of interest is homogenized, eventually condensing the group structure. Then the cross-sections are recovered to be transferred to INSTANT. The boundary fuel blocks in INSTANT will have cross-sections coming from supercell models, whereas the central fuel blocks remain calculated using the single fuel block model of DRAGON. Indeed, building a supercell for them with only fuel neighbouring blocks should deliver the same result as the reflective fuel block because all neighbours are copies of the central fuel block of interest.

The tracking module used to do the calculation is critical. The use of SYBILT: decreases the calculation time by a factor of 10 compared to EXCELT:. However, the interface current method in SYBILT: may not be accurate for such a large domain due to the DP_N approx-

imation made on hexagonal surfaces. Therefore, it was decided to use `EXCELT:` as well, and a convergence study has been performed with both tracking modules to determine the appropriate parameters. Results will be presented for both modules in chapter 4.

`EXCELT:` enables the use of two types of solving methods for the transport equation: the method of characteristics (MOC) or the collision probability (PIJ). The limitation in the addressing system prevents the use of the PIJ method, but the MOC solver is working with iterative techniques and is supposed to deliver more accurate results especially when there is anisotropy. Graphite leads to rather isotropic scattering, so that if one is using a 64-bits version of DRAGON, a switch from MOC to PIJ might be interesting to save time.

3.6.2 SPH homogenisation

The SPH-homogenisation technique has also been tried on the supercell calculations, and will be compared to the direct homogenisation in chapter 4. This SPH-technique was not tried for single block calculations because in those geometries, a complete homogenisation of the lattice was done. Therefore, all the SPH factors would be equal to 1. On the contrary, in the supercell cases, only the central block of the lattice is homogenised and stored for the further core calculations. In this case, the SPH-technique may have an impact on the homogenised cross-sections and improve the results for the supercell path at the core level apart from the number of groups used in INSTANT.

The best way to perform an SPH-homogenisation would be to store the reactions rates produced by the DRAGON calculation on the detailed geometry and fine energy mesh, and then to perform the same calculation using INSTANT on a supercell with the homogenised condensed cross-sections. However, such a solution was not feasible for two reasons: the geometry of the supercells was not reproducible in INSTANT, and no SPH capabilities were available in the code. Therefore, a transport-transport equivalence has been done with DRAGON in both calculations: the heterogeneous calculation and the homogeneous one. No external script was needed to do so, because the SPH technique is already in place in DRAGON.

The mathematical description of this method is described in 2.11.2. The next paragraph describes how it is done in DRAGON. After the first heterogeneous flux calculation, each block is homogenised separately to produce a set of homogenised and condensed cross-sections representative of each block. The reaction rates belonging to each block are stored as a reference. A homogeneous geometry of the supercell is provided in the edition module of DRAGON. This geometry is exactly similar to the heterogeneous geometry, except that each cell of each block is filled with its homogenised representative set of cross-sections previously recovered from the heterogeneous flux calculation. A transport flux calculation is done in DRAGON on the homogeneous geometry, so that new homogeneous reaction rates

are produced. SPH factors are computed to correct the homogeneous cross-sections stored at the beginning so that the new reaction rates of each block are forced to be equal to the reference reaction rates. The process is iterative, and stops when the SPH factors are converged.

The homogeneous cross-sections are directly corrected by the SPH factors, so that the same scripts could be used to transfer the cross-sections to INSTANT. When an SPH-homogenisation was done, standard parameters of DRAGON were kept in the code. Most of the SPH calculations required between 50 and 150 SPH iterations to converge. For 9 groups, DRAGON encountered a numerical issue that was not completely understood. However, an asymptotic normalisation of the SPH factors in the regions of the reflector decreased the instabilities and ensured the convergence. This feature was not used for other group condensations. The convergence of the SPH factors is meant to be achieved when their variations are lower than $1,0.10^{-4}$.

3.6.3 Convergence study for a supercell model with 2 steps

It is assumed that the supercells are similar enough so that all that is necessary is a convergence study over just one type of supercell. The supercell where the block of interest is surrounded by three reflector blocks (see figure 3.2(c)) was selected. A MCNP calculation was performed for this supercell. The reference K_{eff} obtained through MCNP was:

$$K_{\text{eff MCNP SupCell}} = 1,30955 \pm 0,00011. \tag{3.11}$$

The supercell model was built with homogeneous fuel pellets surrounded by graphite in the fuel cells, and homogeneous hexagonal cells for the others (coolant and graphite cells). A white reflective boundary condition was set. All cross-sections for the supercell calculations were first generated by DRAGON using the single block model (MOC with converged parameters) described above in section 3.4.2 page 68.

`EXCELT:` - Method of characteristics

The flux calculation is first studied with the MOC using modules `EXCELT:` and MCCGT. The MOC enables the use of iterative techniques instead of computing a full matrix, as required with the PIJ method. The PIJ matrix is usually too large to store in memory: DRAGON encounters addressing problems. However, it is likely that a converged mesh for MOC will be suitable for a calculation using PIJs. Thus, a mesh which showed to be discretized enough for one method may also be adequate for the other. Furthermore, it is very likely that the number of groups may not impact the precision of the tracking. Therefore,

the convergence study will be done starting with 26 groups in order to save time. It is assumed that the mesh which will show convergence in 26 groups will also be converged for a calculation with 295 groups.

`EXCELT:` enables the user to define the density of lines over the geometry and the number of angles. It is also possible to discretize and sectorize the cells.

The first set of calculations will study the density of lines. The number of angles will be kept at 4 angles to minimize the calculation time. Moreover, the anisotropy of this reactor is low, which leads to good results even with a small number of angles. The geometry is not discretized nor sectorized. Results are given in table 3.22(a). It was concluded that a density of 10 lines per cm is enough to converge on this parameter.

Table 3.22 – Convergence study for the supercell: tracking parameters

(a) Density of lines

TISO		K-eff
Nb. angles	Density (lines.cm^{-1})	
4	500	1,31283
4	400	1,31283
4	300	1,31283
4	200	1,31284
4	100	1,31283
4	70	1,31283
4	40	1,31283
4	30	1,31283
4	20	1,31283
4	10	1,31282

(b) Number of angles

TISO		K-eff
Nb. angles	Density (lines.cm^{-1})	
36	20	1,31286
30	20	1,31285
24	20	1,31285
18	20	1,31284
12	20	1,31281
10	20	1,31286
8	20	1,31285
6	20	1,31255
4	20	1,31283
2	20	1,31255

Table 3.22(b) records the results for different numbers of angles but a constant density of lines set to 20 lines.cm^{-1}. The other parameters are the same as above. It was concluded that 4 angles are enough to describe this geometry. This small number is due to the fact that the graphite is rather isotropic because of its high mass compared to the neutrons. On the contrary, if the moderator were water, the hydrogen would have a mass near to the one of the neutrons, which would introduce more anisotropy in the calculation and probably necessitate a finer angular discretization.

Finally, some calculations were performed to see if further discretization of the geometry would significantly change the K_{eff}. For those calculations, 8 angles and 70 lines per cm were used. The geometry was sectorized and radially discretized in the fuel and then in the other types of cells. Table 3.23(a) gives the results. From this, one sees that the radial discretization of the fuel pellet is not of interest, but the sectorization introduces about 58

pcm of difference, which is not negligible. Therefore, in the final model, the sectorization of the fuel cells is retained.

Table 3.23 – Convergence study for the supercell: fuel and graphite cell discretization and sectorization

(a) Fuel cells

Geometry		K-eff
Number of radii	Sectorization	
9	yes	1,31210
6	yes	1,31211
5	yes	1,31210
1	yes	1,31209
1	no	1,31285

(b) Graphite cells

Geometry		K-eff
Number of radii	Sectorization	
3	yes	1,30843
2	yes	1,30843
1	yes	1,30850
1	no	1,31210

3 radii were kept in the fuel pellet and the sectorization was performed into 6 triangles to study the effect of discretization in the surrounding graphite and coolant cells. Results are given in table 3.23(b). This shows that there is a very significant effect of the sectorization in the graphite cells. Fortunately, they do not require fine radially discretization, but the sectorization complicates the input decks and above all, it necessitates more memory resources and it slows down the calculation.

In conclusion, the following settings were adopted for the supercells using `EXCELT:` and the method of characteristics:

- Tracking parameters: TISO, 4 angles, 10 lines.cm^{-1}
- Fuel cells: no discretization of the pellet: radii 0,0 0,6225. Sectorization into 6 triangles.
- Graphite and coolant cells: 1 radius at 0.4 cm and sectorization.

With this configuration and 26 groups, the following is obtained:

$$K_{\text{eff SC MOC 26gr}} = 1,30854 \tag{3.12}$$

which is at -77 pcm from MCNP. The external convergence is reached after 44 iterations. A complete calculation with the generation of the cross-section files takes a bit less than 3 hours on a single processor.

With 295 groups, the following was obtained:

$$K_{\text{eff SC MOC 295gr}} = 1,30807 \tag{3.13}$$

which is at -113 pcm of MCNP, that is to say the same order of magnitude. The calculation

times differ greatly: 2h39min with 26 groups compared to 82h10min with 295 groups (e.g. 3,5 days).

The other two types of supercell were built with the same parameters.

`SYBILT:` - method of collision probabilities

The calculation on supercells were also performed with the `SYBILT:` tracking module. This module uses the collision probability method and the interface currents method so that the calculation time is greatly decreased. However, the module is usually not trusted on large lattices, and therefore it has to be compared to another one.

A convergence study was performed following the same scheme as for `EXCELT:`, so extensive explanation is not repeated here. The feature DP01 prevented the code from converging, so all cases set a DP00 condition, that is to say an isotropic distribution of the current at the interfaces. Table 3.24 gives the different Eigenvalues obtained when varying the integration parameters. QUA2 4 8 (4 angles, 8 basis points) was kept for the next calculations.

Table 3.24 – Convergence study: number of angles and basis points

QUA2		K-eff
Nb of angles	Nb of segments	
8	10	1,28670
6	10	1,28671
4	10	1,28670
3	10	1,28671
2	10	1,31283
8	10	1,28670
8	8	1,28671
8	6	1,28670
8	4	1,28666
8	3	1,28665

The discretization of the geometry on fuel and graphite cells was then investigated (tables 3.25).

Table 3.25(a) shows that the radial discretization is not important in the fuel cells, but that the sectorization has a strong impact on the K_{eff}. The sectorization with no radial discretization was kept for the fuel cells in the following calculations which evaluate different configurations for the other types of cells (table 3.25(b)).

For the graphite or coolant cells, the sectorization proved to be important as well, and so did the radial discretization to a lesser extent. Those features will be kept in the final model.

Table 3.25 – Supercell: discretization, sectorization of the graphite and coolant cells

(a) Fuel cells

Geometry		K-eff
Number of radii	Sectorization	
6	yes	1,28672
3	yes	1,28672
1	yes	1,28671
1	no	1,28798

(b) Graphite and coolant cells

Geometry		K-eff
Number of radii	Sectorization	
3	yes	1,28651
1	yes	1,28671
1	no	1,29134
0	no	1,29137

Again, this will significantly slow down the calculations, as was the case with the `EXCELT:` tracking method.

In conclusion, the following model was kept for all supercells which use the `SYBILT:` tracking module:

- Tracking parameters: QUA2 4 6
- Fuel cells: no discretization of the pellet: radii 0,0 0,6225 (cm). Sectorization into 6 triangles.
- Graphite and coolant cells: 3 radiii at 0,0 0,207 0,415 0,6225 (cm) and sectorization into 6 triangles.

With this configuration and 26 groups, the following is obtained:

$$K_{\text{eff SC SYB 26gr}} = 1,28650 \tag{3.14}$$

which is at -1760 pcm compared to the MCNP result. The external convergence is reached after 26 iterations. A complete calculation with the generation of the cross-sections files takes a bit less than 7 minutes in 26 groups on a single processor.

With 295 groups, it takes about 8 hours and the following was obtained:

$$K_{\text{eff SC SYB 295gr}} = 1,28544. \tag{3.15}$$

This is at -1841 pcm of the MCNP Eigenvalue. Curiously the error seems to be higher with more groups, but it remains in the same order of magnitude.

Notice here that the `SYBILT:` solver is much faster than the MOC solver. In 26 groups, performing the calculation with the method of characteristics takes already too much time, not speaking of the week needed to perform it in 295 groups. However, the `SYBILT:` solver seems to be less accurate when comparing its Eigenvalue to the MCNP results. It will be evaluated whether this poor lattice solution really affects the cross-section generation and to

what extent compared to the method of characteristics, which appears to be more reliable.

In the future, the MOC solver may be improved by some parallelization of the code and better acceleration techniques. This, however, is outside the scope of this work.

3.7 Conclusion for Chapter 3

Two main calculation schemes were presented in this study: the single block path and the supercell path. Both paths share the same core calculation scheme in INSTANT in order to isolate the effect of the cross-section preparation on the core calculation. A comparison with MCNP enable one to evaluate the precision of each scheme according to the number of groups used in the core calculation and the type of cross-section generation. MCNP was also used as a reference to validate the lattice calculations.

A convergence study was performed in INSTANT and shows that a P_3 calculation is converged with a high order on the source ($p_{source} = 5$) and on the interior polynomial expansion ($p_{int} = 6$), while a lower order can be used for the surface polynomial expansions ($p_{surf} = 2$). In addition, no difference was noticed on the Eigenvalue between a P_3 and a P_5 calculation, which indicate that the convergence is achieved with P_3. A linear anisotropic scattering was chosen because the difference with isotropic scattering was not negligible, even if the core seems to be rather diffusive due to the graphite.

Convergence studies were also pursued in DRAGON in order to choose the main parameters and solvers to be used. Those convergence studies mainly show that:

- The white reflective condition applied in DRAGON may not introduce an important bias because the lattices are large enough to ensure the equivalence with a true reflective condition.
- The self-shielding is not strongly affected by the surrounding graphite density. This was unexpected, but fortunately it enable one to perform supercell calculations from homogeneous fuel pellets. In this calculation scheme, the self-shielding is performed during the generation of the homogeneous cross-sections with a single block calculation.
- The real boundary of a single block has to be simulated by an additional row of hexagons in DRAGON, with special densities. The establishment of those densities is described in Appendices C and D.
- The single block calculation can be performed with both `SYBILT:` or `EXCELT:`/`MCCGT:` solvers. Both produce an Eigenvalue that is within ± 60 pcm of the MCNP lattice reference.
- The supercell path is made up of three different supercells with one row of neighbouring blocks surrounding the block of interest. The neighbours are fuel or reflector blocks, de-

pending on the location. While the MOC calculations show a very good agreement with MCNP (about 100 pcm), the `SYBILT:` solver lead to higher errors(about 1800 pcm). Cases starting with only 26 groups in DRAGON for the supercell calculations are also used because it may be a way to accelerate the calculation schemes.

– The reflector calculation is performed with the `SNT::` solver on an annular approximation of the core. Despite its simplicity, this model leads to a good agreement with the $1/12^{\text{th}}$ MCNP core model. The same reflector cross-sections are used for all core calculation so that the effect of the fuel cross-sections in the core calculation is isolated.

CHAPTER 4

RESULTS & DISCUSSION ON A CALCULATION SCHEME

4.1 Reference MCNP calculation

In MCNP, the K_{eff} for a $1/12^{th}$ core calculation with a void boundary condition is $1,24963 \pm 0,00010$. This will be used to evaluate other solutions coming from INSTANT calculations, along with the average fission rates over each block. These give a measure of the accuracy of the shape of the flux and of the cross-sections. This point is of capital importance from an industrial point of view for predicting the power peaks as well as the depletion of the fuel [7].

All calculations were performed on the high-performance computer HELIOS of the INL using a single node with 4 Intel® Xeon processors for each calculation. Each processor has a frequency of 2,66 GHz. Once selected, the node is reserved for the calculation, so that each input is running alone.

4.2 1^{st} path: cross-sections from single-block

In this first section, results are presented of INSTANT calculations obtained with cross-sections generated by the single block path in DRAGON. A calculation is considered good when the comparison with MCNP shows less than 150 pcm of deviation on the Eigenvalue.

4.2.1 MOC path: comparison of the Eigenvalue

Single block, double heterogeneity, 295 groups, MOC

A first set of core calculations was performed where the fuel cross-sections were generated by the single block model computed from scratch (including compacts containing double heterogeneity, with 295 energy groups, abbreviated by DH295) with the method of characteristics. In most cases, the homogenized cross-sections were then condensed to a coarse energy structure. Table 4.1 summarizes the results for every group-structure used in INSTANT whole core calculations.

The deviation is calculated with the following formula:

$$\text{Deviation} = \frac{K_{\text{eff INSTANT}} - K_{\text{eff MCNP}}}{K_{\text{eff MCNP}}} \cdot 10^5 \text{ (pcm)} \tag{4.1}$$

Table 4.1 – Single block, case MOC DH295: Eigenvalues

Nb of groups in INSTANT	SB MOC DH295	
	P_3 scat 1	
	Core K-eff	Deviation from MCNP (pcm)
295	1,25010	38
26	1,25023	48
23	1,24970	6
12	1,24968	4
10	1,24987	19
9	1,24660	-242
6	1,24758	-164
4	1,20891	-3259
2	1,21162	-3042

First, it can be seen that when a sufficiently high number of groups is used in the whole core calculation, the agreement between deterministic calculation and MCNP is quite good. This is especially the case between 295 groups and 10 groups.

These results are very encouraging, because they show that keeping a quite large number of groups in the whole core calculation may help to reduce the errors created by a lattice calculation which does not take into account the environment. The rather inaccurate neutron energy spectrum used to homogenize cross-sections over the single block may be partially compensated for by keeping a large number of groups, so that the neutron energies may be redistributed during the core calculation, thus accounting for neighbouring reflector blocks which were not present in the lattice calculation.

It can also be observed that keeping 26 groups at the core level is enough to reproduce the results obtained with 295 groups, with a discrepancy as small as 10 pcm. This shows that there is no need to perform the core calculation with 295 groups if the single block path is used. It does not provide such an improvement which would indicate a better modelling of the physics in the core. This statement is valid at this point for the determination of the Eigenvalue only. A study of the fission maps is still in order to complete this statement.

With 9 groups and fewer in INSTANT, the results are becoming worse, or even completely wrong with 4 or 2 groups. The definition of the group boundaries may explain that better results are achieved with 6 groups than with 9 groups. Compensation of errors can also give fortuitous results in Eigenvalue.

On a more general consideration, the establishment of the boundaries for the groups seems to have a large influence, as the accuracy seems to be improved with 23 groups compared to 26. This may be due to more appropriate boundaries of the 23-group structure. A deeper

analysis should be performed to confirm this statement and an optimized group structure may be adapted to this fuel.

Single block, homogeneous hexagons, 295 groups, MOC

A second set of core calculations were performed where the fuel cross-sections had been generated by the single block model computed using a previous single block calculation which gives homogeneous cross-sections for the individual hexagonal cells in 295 groups (abbreviated by HOM295). This was done to provide a fair comparison to the supercell paths. Again, MOC was used for the DRAGON calculations.

Table 4.2 summarizes the results for every group-structure. They are very similar to those computed from the original detailed block model. It means that the first single block calculation was well suited to carry the double-heterogeneity effect through the homogenization. It is encouraging to see this result, because it indicates that the supercell calculations will be started with a good set of cross-sections. Indeed, the same initial set of cross-sections will be used for this case.

Table 4.2 – Single block, case MOC HOM295: Eigenvalues

Nb of groups in INSTANT	SB MOC HOM295 P3 scat 1	
	Core K-eff	Deviation from MCNP (pcm)
295	1,24971	6
26	1,24977	11
23	1,24947	-13
12	1,24926	-30
10	1,24947	-13
9	1,24625	-270
6	1,24716	-198
4	1,20861	-3283
2	1,21094	-3096

The agreement with MCNP is essentially equivalent to the previous results given that some discrepancy is expected between MCNP and a deterministic code. It shows that both ways are relevant and that it will be worthwhile to analyse the fission maps of the supercell paths.

Single block, homogeneous hexagons, 26 groups, MOC

Here, to generate homogenized cross-sections for INSTANT, a single block calculation was performed using homogeneous cells, but this time starting with 26 groups. It means that a first detailed single block calculation is performed in 295 groups to produce homogenized cross-sections for each type of elemental hexagonal cells, but those cross-sections are condensed to 26 energy groups instead of keeping the initial 295-group structure. Then a single block calculation with homogeneous elemental hexagonal cells is performed with 26 groups, producing homogenized cross-sections for the entire fuel block which are used in the INSTANT whole core calculation. The cross-sections may be condensed to a coarser energy group structures (with less than 26 energy groups) before doing the whole core calculation. The MOC solver was used in DRAGON.

This has been done for two reasons:

- the supercell calculations starting with 26 groups are much faster. A fair comparison was needed for the single block paths in order to estimate the added value of carrying a better spectrum at the lattice level.
- it provides an estimation of the influence of the number of groups taken at the lattice level for a standard fuel block.

Table 4.3 summarizes the results. Fewer type of group structures could be used because starting with 26 groups, the possibilities of choosing new boundaries are limited by the boundaries of the 26-group structure. It happens that the 23-, 13- and 9-group structures could not be produced because some of their epithermal groups were located inside one large energy group of the 26-group structure. As it is not straightforward to unfold an energy group into smaller intervals, the condensation was not performed for them.

The 10-group structure used in these calculations was created from the 12-group structure to take into account this problem, but it is then no more a reference. well established group-structure. Other group-structure were left as they were. Nevertheless the 26 group calculation is here probably the most interesting of them, because the run time in INSTANT is already very manageable.

One here observes the same errors appearing when using fewer than 10 groups in INSTANT. The results with 26 or 10 groups are quite comparable. They both lead to a higher Eigenvalue than the previous calculations, but it remains acceptable. It is shown here that conducting a lattice calculation with 26 groups instead of 295 may be conceivable if it is necessary due to time constraints. Otherwise, it is of course preferable to conserve 295 groups at the lattice level.

Table 4.3 – Single block, case MOC HOM26: Eigenvalues

Nb of groups in INSTANT	SB MOC HOM26	
	P_3 scat 1	
	Core K-eff	Deviation from MCNP (pcm)
26	1,25060	78
10	1,25070	86
6	1,24743	-176
4	1,21207	-3006
2	1,21211	-3002

4.2.2 `SYBILT:` path: comparison of the Eigenvalue

In this section, all calculations over the single blocks were performed using the `SYBILT:` solver, that is to say a collision probability method and the interface current coupling method. When a previous calculation was necessary to perform a pre-homogenisation, the same set of cross-sections for the MOC calculation was used, in order to be consistent.

The same abbreviations have been used, so they will not be repeated here.

Single block, double heterogeneity, 295 groups, `SYBILT:`

Table 4.4 presents the results using directly the double-heterogeneity model.

Table 4.4 – Single block, case SYB DH295: Eigenvalues

Nb of groups in INSTANT	SB SYB DH295	
	P_3 scat 1	
	Core K-eff	Deviation from MCNP (pcm)
295	1,25105	114
26	1,25114	121
23	1,25084	97
12	1,25069	85
10	1,25089	101
9	1,24764	-159
6	1,24863	-80
4	1,20993	-3177
2	1,21270	-2955

On the one hand, the `SYBILT:` solver gives clearly worse results than the method of characteristics, but the Eigenvalues are reasonably good. The error is about 100 to 140 pcm

with MCNP, which is around a factor of two larger than the errors obtained with the MOC. On the other hand, the SYBILT: solver is faster than the MOC.

Single block, homogeneous hexagons, 295 groups, SYBILT:

As mentioned previously, the cross-sections used to perform the homogeneous calculations in DRAGON were recovered from a heterogeneous single block calculation with MOC, so that subsequent DRAGON calculation are started with the same set of cross-sections.

Table 4.5 summarizes the results for every group-structure.

Table 4.5 – Single block, case SYB HOM295: Eigenvalues

Number of groups	SB SYB HOM295	
	P_3 scat 1	
	Core K-eff	Deviation from MCNP (pcm)
295	1,25024	49
26	1,25031	54
23	1,24994	25
12	1,24993	24
10	1,25011	38
9	1,24691	-218
6	1,24776	-150
4	1,20897	-3254
2	1,21184	-3024

Here it is interesting to note that the Eigenvalues of the better group-structures (from 10 to 295 groups) agree well with MCNP. This may come from the fact that the SYBILT: calculation began with a set of cross-sections that have been generated by the MOC. The double-heterogeneity treatment may be better handled by the MOC solver, and the calculation with homogeneous cells is easier for SYBILT:. It would explain the fact that we see a better agreement between the MOC and the SYBILT: calculations.

Again, calculations with fewer than 10 groups appear irrelevant.

Single block, homogeneous hexagons, 26 groups, SYBILT:

Finally, table 4.6 shows the results for the available group-structures starting with 26 groups and homogeneous cells. The same set of cross-sections as for the MOC method were used.

The results are quite similar to those obtained with the MOC. The same explanation is proposed: cross-sections were prepared by an MOC solution, so that the double-heterogeneity

Table 4.6 – Single block, case SYB HOM26: Eigenvalues

Nb of groups in INSTANT	SB SYB HOM26	
	P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25074	89
10	1,25099	109
6	1,24765	-158
4	1,21241	-2978
2	1,21243	-2977

treatment is performed with an accurate solver, and then the calculation over homogeneous cells leads to a good agreement with MCNP.

Compared to MCNP, there is an error of about 90 to 110 pcm in the best cases (26 and 10 groups).

4.2.3 Fission rate maps

General overview of the fission rate maps for the 1st path

In order to compare various whole core solvers, power shape is an important metric. INSTANT; however, does not contain information on energy per fission. Therefore, in this work, fission rates were used for comparison to MCNP.

It is not possible to plot all fission maps in all cases in this thesis. It was decided to present the fission maps obtained from the single block case starting with double heterogeneity and without condensation in energy groups, so that the INSTANT calculation is performed with 295 groups. The P_3 calculation was used in INSTANT. Those choices were driven by the consideration that this case should theoretically be the best case of all the calculations using the single block path.

Figure (4.1) shows a detailed fission rate map of the core. Starting from the center, there are first blocks of reflector containing no fissile elements and therefore no fissions. Then the first inner ring of fuel blocks is encountered where the fission rate reaches a maximum and then decreases. The gradient inside this block is very important, and presents a modelling challenge. Though in an actual design, this power peak would be suppressed with burnable poisons, the thermal flux peak still needs to be accurately resolved in any case.

The fission peak is explained by the high number of thermal neutrons coming from the reflector which are quickly absorbed inside the first few cells containing fuel. Because the fuel surrounds the inner reflector, the thermalization of neutrons is very efficient because

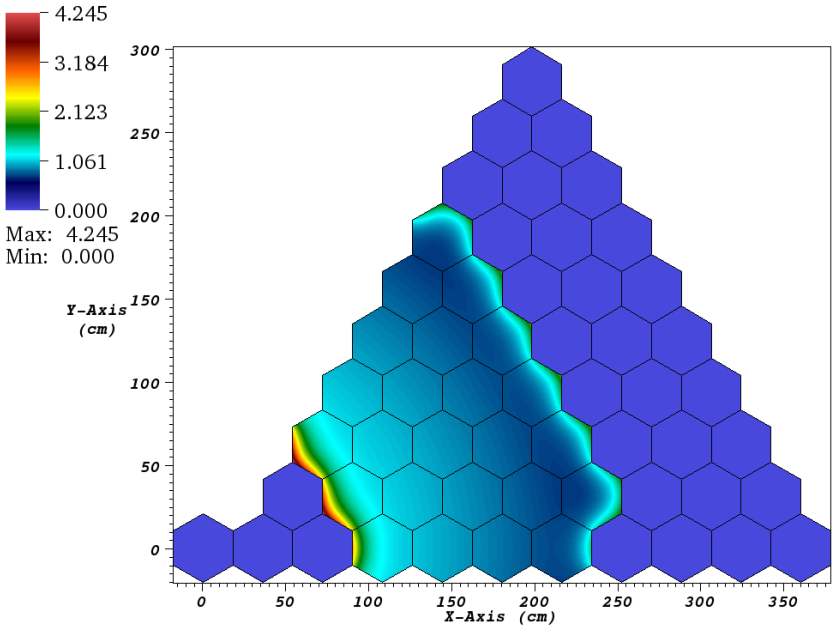
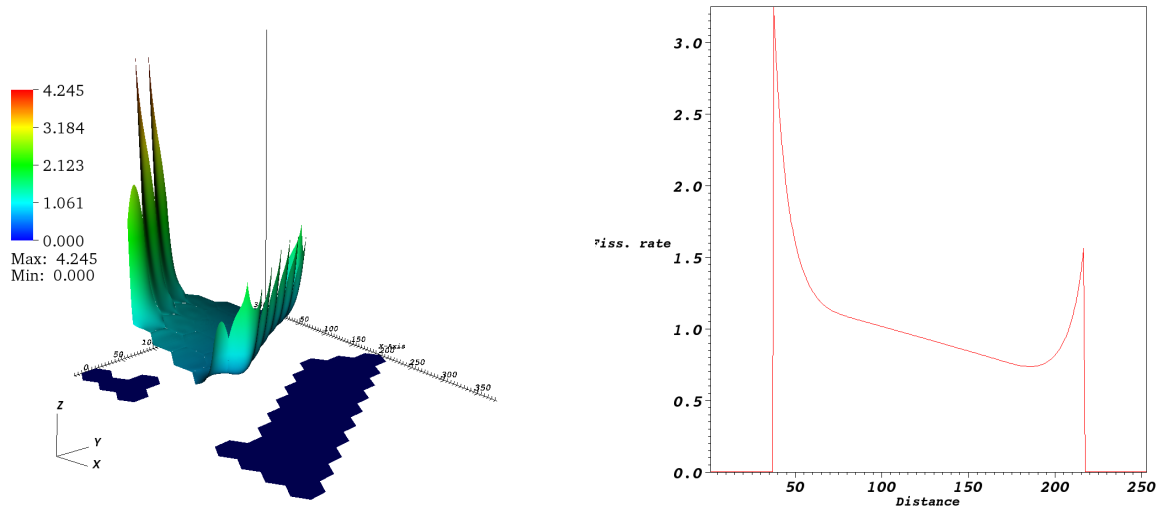


Figure 4.1 – Map of the detailed fission rates in the core - SB, 295 groups

leakage is not likely to occur. Almost all fast neutrons emitted by the core and entering the inner reflector are well thermalized and then reabsorbed in the innermost fuel adjacent to the central reflector.

Moving radially outward from the power peak, the fission rates decrease. The reflector is probably no longer affecting the flux, and a shape that follows approximately the fundamental mode with leakage may be observed. But upon reaching the outer fuel boundary adjacent to the outer reflector, the fission rate rises again and forms a second peak before vanishing at the outer reflector. This peak comes from the neutron thermalization in the outer reflector: some neutrons coming from the fuel enter the reflector, are thermalized, and re-enter the fuel where they are absorbed quickly in the first fuel cells. This peak is smaller because contrary to the inner part, here leakage is more probable. Geometrical effects also matter. Figure (4.2(a)) shows the same data on a 3-dimensional plot, where the fission average is normalized to 1.

It is also possible to plot the fission rate along a line taken across the core, starting inside the reflector, crossing the fuel regions and finishing in the outer reflector. This is shown in



(a) Elevation: the Z-axis represents the fission rate (b) Radial traverse across core

Figure 4.2 – Fission rates using SB model and 295 groups in INSTANT

figure 4.2(b).

In this plot, the power peaks at both fuel/reflector interfaces are well appearing, with the highest power reached at the inner interface. The power in the reflector is null, naturally.

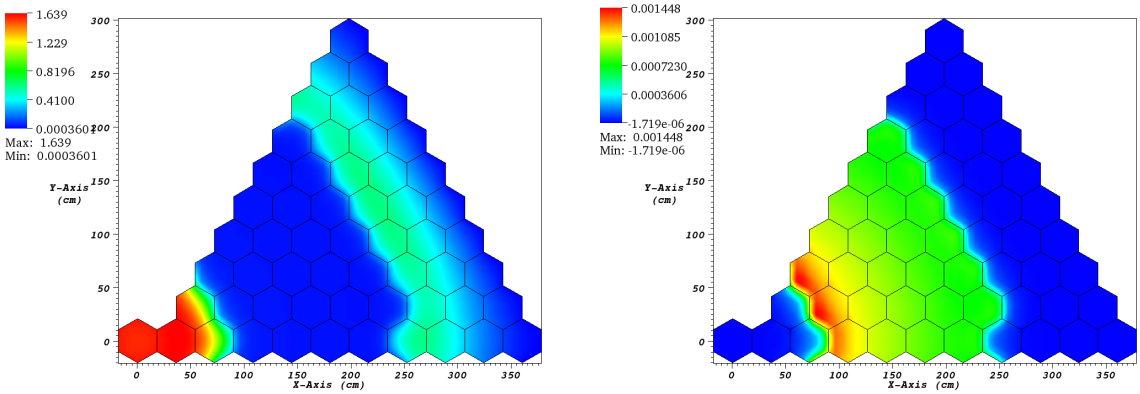
There is a factor of 4,2 between the average fission rate (normalised to 1 here) of the fuel and the maximum fission rate at the inner interface. In an actual design, burnable poisons and control rods will flatten the peak. The goal is to achieve the flattest power distribution as possible while the reactor remains critical. This is important for thermofluid reasons, mechanical concerns to minimize the stress in the TRISO's particles, and also regarding the depletion of the fuel, which we want to be as homogeneous as possible to increase the duration of a campaign and facilitate the refuelling operations such as the treatment and recycle process of the spent fuel. However, the peaks may not disappear completely and the spectral effect will still be present.

This awkward shape of the flux is characteristic of an annular core like those designed for VHTR. The annular design was chosen so that it withstand conduction cool-down.

Figure (4.3(a)) shows the flux for neutrons belonging to group number 295, that is to say the neutrons of lowest energy. The scale is arbitrary.

This shows that the reflector moderates the neutron energy spectrum. As explained above, a larger fission peak is expected in the first row of fuel blocks than in the outer row. Indeed, the flux is lower at the outer interface because of greater leakage.

On the contrary, the neutron flux in group 1 (Fig. 4.3(b), arbitrary scale), where neutrons



(a) Thermal flux in the core, group 295

(b) Fast flux in the core, group 1

Figure 4.3 – Flux analysis: SB MOC DH295, 295 groups

have the highest energies, is logically high in the fuel regions and nearly null in the reflector. We also notice that the peak of the fast flux is located on the inner part of the fuel. This is because of the fission rate is higher in the inner part of the fuel, generating more fast neutrons in this region.

Finally, a fission rate map is shown where the fission rates have been averaged over each hexagonal block (4.4(a)). This will allow comparison with block-averaged fission rate tallies from MCNP.

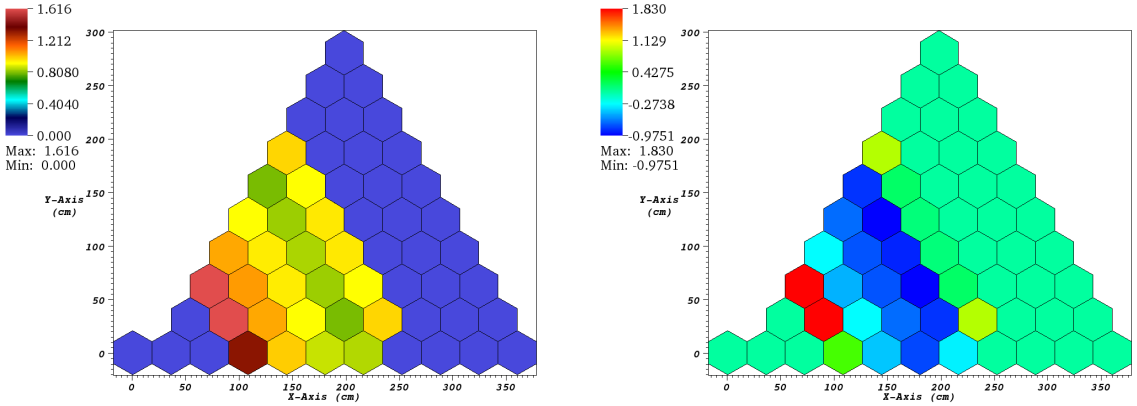
In these plots, the fission rates have been normalised such that the average of the fission rates over all fuel assemblies in the core is equal to 1. Similar power peaking observations can be made. As seen in previous plots, we observe the two thermal peaks in the fuel regions adjacent to the reflectors, with a higher peak in the inner part.

The formula used to generate the errors is in each block is given by:

$$\text{Error} = \frac{\text{Fission rate in INSTANT} - \text{Fission rate in MCNP}}{\text{Fission rate in MCNP}} \times 100(\%) \tag{4.2}$$

Figure 4.4(b) shows the differences observed between the case SB MOC DH295 295 groups with the MCNP reference. Note that the largest statistical uncertainty on the fission rate tallies in MCNP is worth 9.10^{-3}, so about 1 %.

From this, it can be observed that the inner regions of the fuel is under-predicting fission rates relative to the MCNP results, while it over-predicts the fission rate in the innermost



(a) Block-averaged fission rates

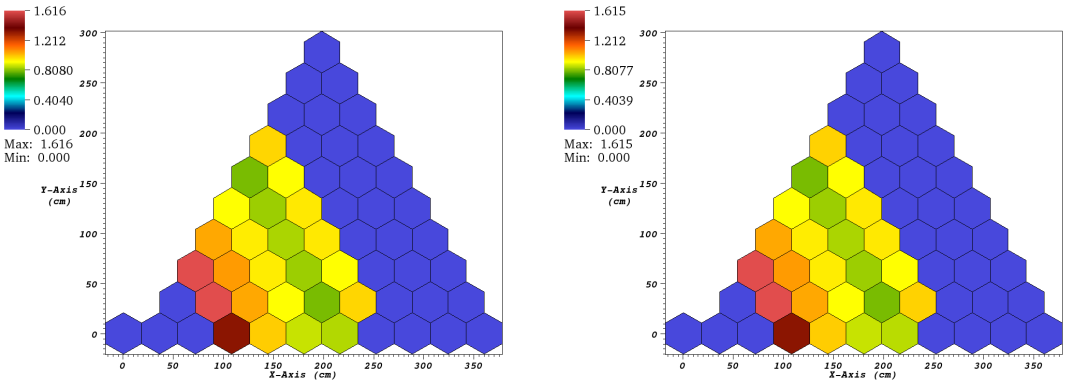
(b) Percent error on block-averaged fission rate

Figure 4.4 – SB MOC DH295, 295 groups - Block-averaged fission rates and associated error compared to the MCNP reference calculation

ring of fuel at the inner interface. There is an over-estimation of 1,83 % at the inner fuel reflector interface, and an under-estimation of -0,98 % in the middle-right of the core. In average, we have -0,14 % of errors with MCNP. The standard deviation is equal to 0,81 %, which is quite large compared to the average value. This is due to the high magnitude of the errors in the inner peripheral blocks.

Figures 4.5, 4.6 and 4.7 give the same types of maps for different group condensations. With 6 or more groups, the global aspect of the fission maps remains the same with the presence of two peaks in the blocks surrounded by reflector. With 4 or 2 groups, this shape totally disappears.

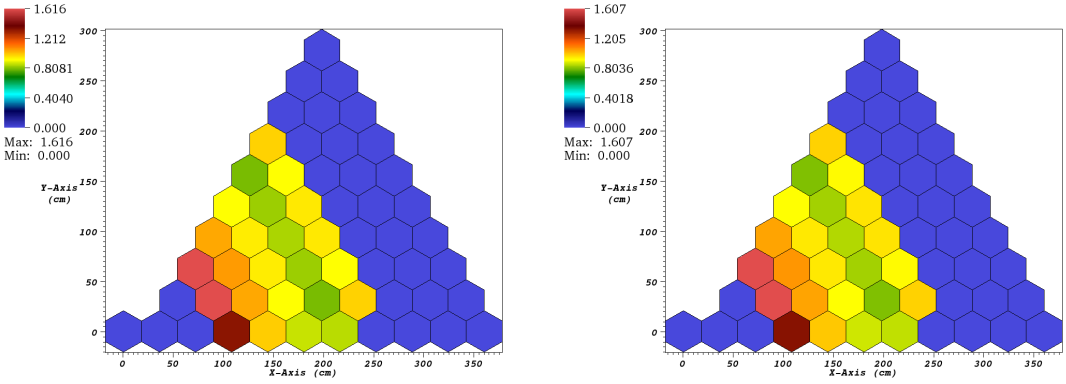
Figures 4.8, 4.9 and 4.10 show the results of the comparison with MCNP. It appears more clearly that the 4 and 2 group structures give a poor shape for the power calculation in the fuel blocks. In the next section, summary information will be presented on the fission maps for each type of calculation in DRAGON, each type of calculation in INSTANT and each condensation (see section 4.2.3).



(a) SB, 295 groups

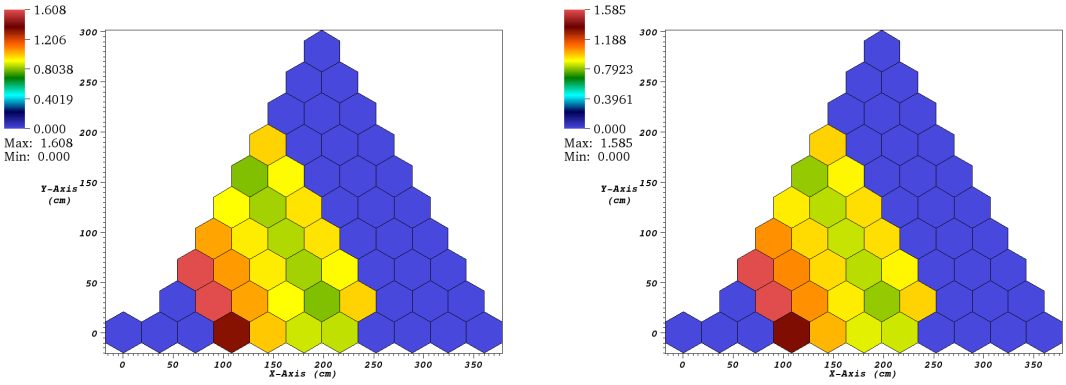
(b) SB, 26 groups

Figure 4.5 – Fission maps for various group condensations in INSTANT, SB MOC DH295



(a) SB, 23 groups

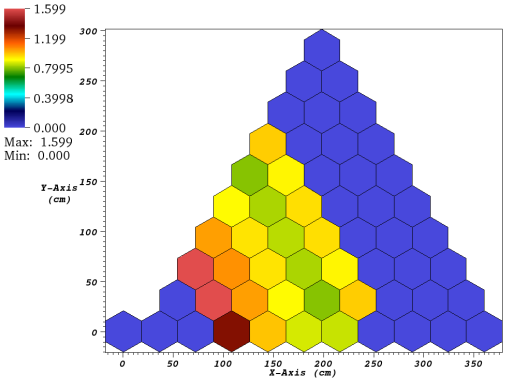
(b) SB, 12 groups



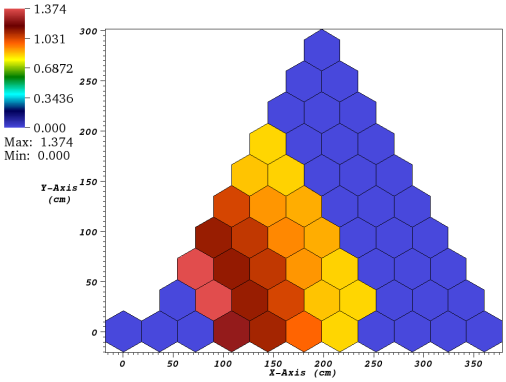
(c) SB, 10 groups

(d) SB, 9 groups

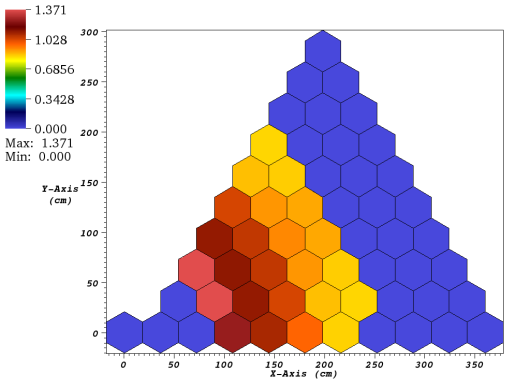
Figure 4.6 – Fission maps for various group condensations in INSTANT, SB MOC DH295 (continued)



(a) SB, 6 groups

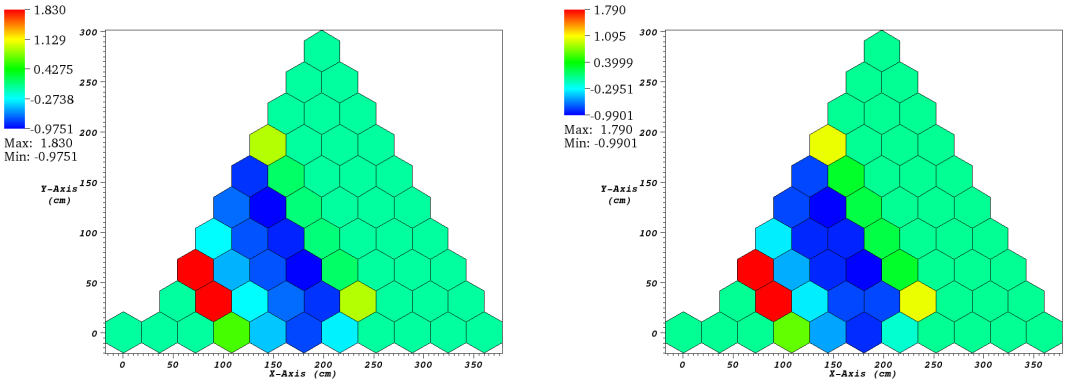


(b) SB, 4 groups



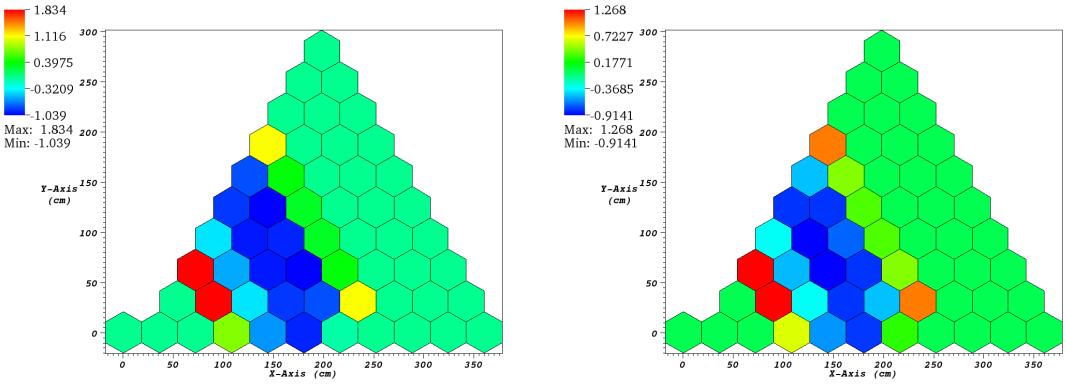
(c) SB, 2 groups

Figure 4.7 – Fission maps for various group condensations in INSTANT, SB MOC DH295 (end)



(a) SB, 295 groups

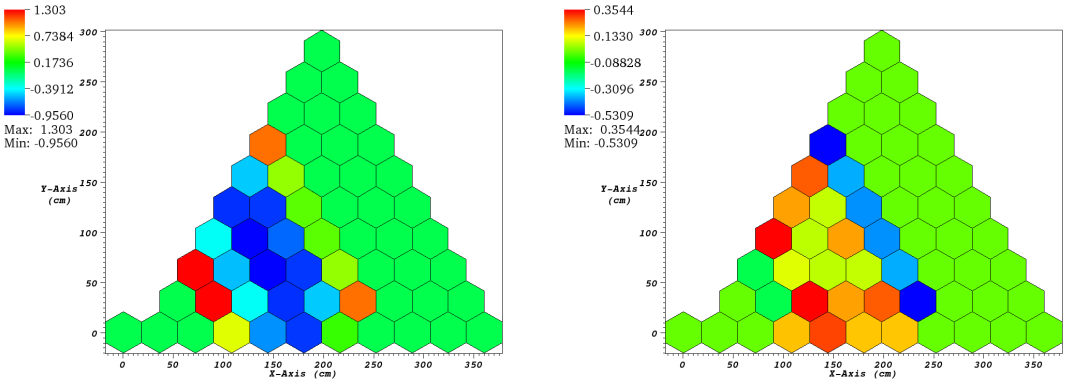
(b) SB, 26 groups



(c) SB, 23 groups

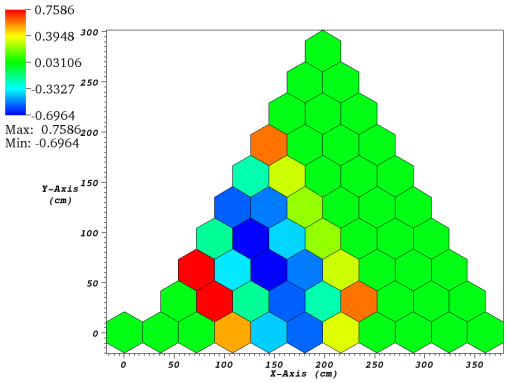
(d) SB, 12 groups

Figure 4.8 – Block-averaged percent deviation from MCNP fission rate for various group structures in INSTANT, SB MOC DH295



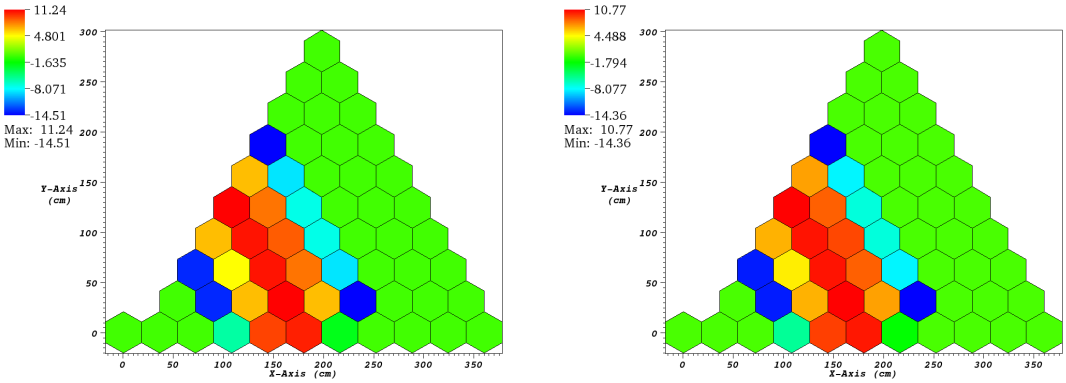
(a) SB, 10 groups

(b) SB, 9 groups



(c) SB, 6 groups

Figure 4.9 – Block-averaged percent deviation from MCNP fission rate for various group structures in INSTANT, SB MOC DH295 (continued)



(a) SB, 4 groups

(b) SB, 2 groups

Figure 4.10 – Block-averaged percent deviation from MCNP fission rate for various group structures in INSTANT, SB MOC DH295 (end)

Extensive study of the fission maps

The following tables summarize the fission rates obtained through the single block paths with maximal positive errors (Max), maximal negative errors (Min), average of the errors (Av. μ), standard deviation of errors (Std σ) and the absolute deviation (AbsD) which is the average of the absolute errors.

Single block, MOC models: Table 4.7 shows the results using MOC and the detailed double heterogeneity. Curiously, the 9-group structure gives the best results compared to MCNP with errors in the interval $[-0,3\%; +0,24\%]$, a small average error and a small standard deviation or absolute average deviation. This unexpected result does not appear consistent with the K_{eff} analysis shown previously, where the 9-group structure lead to a K_{eff} much farther from the reference MCNP value.

Table 4.7 – Single block, case MOC DH295: fission rate study

Nb of groups in INSTANT	SB MOC DH295				
	P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD
295	1,83	-0,98	-0,14	0,81	0,69
26	1,79	-0,99	-0,13	0,85	0,74
23	1,83	-1,04	-0,14	0,90	0,80
12	1,27	-0,91	-0,10	0,73	0,68
10	1,30	-0,96	-0,10	0,76	0,71
9	0,35	-0,53	-0,009	0,48	0,25
6	0,76	-0,70	-0,06	0,29	0,45
4	11,2	-14,5	0,99	9,36	9,01
2	10,8	-14,4	1,01	9,22	8,86

Using 4 or 2 groups is clearly not sufficient to achieve good precision. When the fission rates are plotted, it is clear that the gradients are completely missed, with the maximum reached in the center of the fuel regions instead of the regions adjacent to reflector blocks. The other group-structures match better the MCNP reference with small errors. But, perhaps unexpected, it does not show that the calculation with 295 groups in the whole core calculation is the closest to MCNP regarding the fission rates.

Table 4.8 gives the results for the path using a pre-calculation and the method of characteristics. It is very similar to the previous ones. Indeed, the DRAGON calculations have been done with the same number of groups, and the cross-sections were probably well homogenized by the code before entering the main calculation. Therefore, the resulting cross-sections

passed to INSTANT are very similar to those coming from a direct single block model with all details, and it leads to similar whole-core results.

Table 4.8 – Single block, case MOC HOM295: fission rate study

Nb of groups in INSTANT	SB MOC HOM295				
	P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD
295	1,86	-0,99	-0,14	0,82	0,70
26	1,82	-1,01	-0,14	0,86	0,76
23	1,85	-1,05	-0,14	0,91	0,81
12	1,29	-0,93	-0,10	0,75	0,70
10	1,33	-0,97	-0,10	0,78	0,73
9	0,35	-0,51	-0,007	0,28	0,24
6	0,78	-0,71	-0,06	0,49	0,46
4	11,2	-14,5	0,99	9,46	9,01
2	10,8	-14,4	1,01	9,31	8,86

Again, it is observed that although the K_{eff} matched best between INSTANT and MCNP with larger numbers of groups (295, 26, 23), the fission rate distribution matches best at 9 groups. This is counter-intuitive because more groups should enable one to carry more spectral information and produce more accurate results.

The fission rates remain close to MCNP when the calculations are started with only 26 groups in DRAGON (table 4.9). It seems that this approximation works well.

Table 4.9 – Single block, case MOC HOM26: fission map study

Nb of groups in INSTANT	SB MOC HOM26				
	P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD
26	2,05	-1,03	-0,16	0,87	0,71
10	1,47	-0,91	-0,11	0,71	0,60
6	1,13	-0,69	-0,09	0,55	0,48
4	9,90	-12,7	-0,85	8,27	7,88
2	10,6	-14,2	-0,98	9,19	8,75

Single block, `SYBILT:` models: The same study was produced with cross-sections generated using `SYBILT:` in DRAGON and the single block path. Table 4.10 shows results from the path directly using the double-heterogeneity model, whereas table 4.11 shows results using cross-sections computed in 295 groups with a pre-calculation over a block and table 4.12 give the results for a calculation performed in 26 groups.

The results are very similar to those obtained from cross-sections computed with `EXCELT:`. Curiously, they seem to be even a bit better than the previous results where MOC was used: for example in table 4.10, with 295 groups in INSTANT, the errors are located inside the interval $[-0,94; 1,77]$ whereas with the MOC path (table 4.7, the errors are inside the interval $[-0,98; 1,83]$. So the width of the errors becomes a bit smaller with the `SYBILT:` path. It is difficult to explain that, but the magnitude of the difference is not significant. The conclusion is that regarding the single block model and the resulting fission rate maps in INSTANT, the MOC and the `SYBILT:` solvers are quite equivalent. On the Eigenvalues, the differences and the advantage of the MOC in DRAGON were more apparent.

Table 4.10 – Single block, case SYB DH295: fission rate study

Nb of groups in INSTANT	SB SYB DH295				
	P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD
295	1,77	-0,94	-0,13	0,78	0,66
26	1,73	-0,96	-0,13	0,82	0,72
23	1,76	-1,00	-0,13	0,86	0,77
12	1,20	-0,87	-0,11	0,70	0,65
10	1,24	-0,91	-0,09	0,73	0,68
9	0,37	-0,57	-0,09	0,31	0,28
6	0,69	-0,66	-0,05	0,44	0,41
4	11,2	-14,5	0,99	9,45	9,00
2	10,8	-14,3	1,01	9,30	8,85

Table 4.11 – Single block, case SYB HOM295: fission rate study

Nb of groups in INSTANT	SB SYB HOM295				
	P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD
295	1,83	-0,97	-0,14	0,81	0,68
26	1,79	-0,99	-0,13	0,84	0,74
23	1,83	-1,04	-0,14	0,89	0,80
12	1,26	-0,90	-0,10	0,72	0,67
10	1,30	-0,95	-0,10	0,76	0,71
9	0,36	-0,55	-0,009	0,29	0,26
6	0,75	-0,69	-0,06	0,47	0,44
4	11,2	-14,5	0,99	9,46	9,01
2	10,8	-14,4	1,01	9,32	8,86

Table 4.12 – Single block, case SYB HOM26: fission rate study

Nb of groups in INSTANT	SB SYB HOM26				
	P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD
26	2,04	-1,02	-0,15	0,87	0,71
10	1,46	-0,90	-0,11	0,70	0,59
6	1,12	-0,68	-0,08	0,54	0,47
4	9,90	-12,7	-0,85	8,27	7,88
2	10,6	-14,3	-0,98	9,19	8,75

4.2.4 Conclusion on the single block calculation scheme

It has been shown that the single block path provides fairly good results so long as more than 10 groups are used in the whole-core calculation. The benefit of increasing the number of groups in the core calculation above 12 groups is not obvious from these calculation. In particular, keeping 295 groups is not leading to better results compared to the MCNP reference. The error made on the K_{eff} when using 10 or more groups is typically about 50 pcm approximately. When the cross-sections are prepared in DRAGON starting with 26 groups, it increases this error to approximately 100 pcm. Very good agreement was shown between the calculations starting from scratch or starting with homogeneous cells and 295 groups, and this may be explained by the fact that the double-heterogeneity effect is well carried by the first homogenisation of the fuel pellet in DRAGON.

The results of whole core calculations show little difference between cross-sections generated by the MOC solver or the `SYBILT:` solver. As both methods showed a rather good agreement with MCNP at the lattice level, this point was expected.

4.3 2nd path: cross-sections from supercells

The supercell path has been evaluated with the tracking modules `SYBILT:` and `EXCELT:`/-`MCCGT:` (MOC). A first calculation was performed on a single block to recover homogeneous properties for each cell, because a direct calculation on a fully-detailed supercell required too much memory. The supercell calculations have been performed starting with 26 and 295 groups in DRAGON. Here the direct homogenisation was used. The results of the calculations involving the SPH technique will be grouped together and analysed in a separate section (section 4.3.4).

4.3.1 MOC path: comparison of the Eigenvalues

In this section, results are shown for the cross-sections generated in DRAGON with the method of characteristics. In INSTANT, the calculations were carried out in P_3 with linear anisotropic scattering.

Supercells, homogeneous hexagonal cells, 295 groups, MOC

Table 4.13(a) shows the results for every group-structure when the cross-sections were generated in DRAGON using 295 groups. The corresponding results with the single block path are recalled next to it (table 4.13(b)).

Table 4.13 – Comparison of the Eigenvalues between supercells and single block paths, MOC solver in DRAGON with 295 groups

(a) Supercells, case MOC HOM295

Nb of groups in INSTANT	SC MOC HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24953	-8
26	1,24984	17
23	1,24980	14
12	1,25099	109
10	1,25126	130
9	1,25198	188
6	1,25231	214
4	1,25939	781
2	1,26236	1019

(b) Single block, case MOC HOM295

Nb of groups in INSTANT	SB MOC HOM295 P3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24971	6
26	1,24977	11
23	1,24947	-13
12	1,24926	-30
10	1,24947	-13
9	1,24625	-270
6	1,24716	-198
4	1,20861	-3283
2	1,21094	-3096

It can be observed that the agreement is quite good when using 295 groups for either calculation scheme: only -8 pcm and 6 pcm compared to MCNP. When a condensation was performed before the core calculation, the agreement is still within 150 pcm from 26 to 10 groups are used in INSTANT, and then quite poor with 9 groups and fewer.

With 295, 26 or 23 groups, no clear advantage can then be found for the supercells at this stage regarding the K_{eff}. With 12 and 10 groups, the advantage stands for the single block path. It seems difficult to explain this observation, but it can most likely be attributed to cancellations of errors. With very few groups (2 or 4), the supercell are clearly providing a significant improvement compared to the single block path: for example with 4 groups, 781 pcm of error is obtained with supercells, compared to 3283 pcm of errors with the single block path. However, with 781 pcm of deviation to the reference, the error remains high.

Supercells, homogeneous hexagons, 26 groups, MOC

Table 4.14(a) gives the results for every group-structure when the supercell calculations were performed in DRAGON starting with 26 groups. As usual, the corresponding results are recalled from the single block path shown next to it in table 4.14(b).

Table 4.14 – Comparison of the Eigenvalues between supercells and single block paths, MOC solver in DRAGON with 26 groups

(a) Supercells, case MOC HOM26

Nb of groups in INSTANT	SC MOC HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25003	32
10	1,25118	124
6	1,25273	248
4	1,25832	695
2	1,26230	1014

(b) Single block, case MOC HOM26

Nb of groups in INSTANT	SB MOC HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25060	78
10	1,25070	86
6	1,24743	-176
4	1,21207	-3006
2	1,21211	-3002

As usual, the results are quite good when 26 or 10 groups are kept in INSTANT. With fewer than 10 groups, INSTANT gives poor results. It is also noticed here that the supercell path is providing a significantly better agreement with MCNP when a small number of groups is used in INSTANT (2 or 4 groups). However, the Eigenvalue remains at least at 695 pcm from MCNP with the supercell path and 4 groups.

With 26 or 10 groups, approximately the same Eigenvalues are obtained as when performing this calculation with cross-sections generated with 295 groups in DRAGON (see section 4.3.1). This suggests that the lattice calculation may be done with 26 groups for the supercells instead of carrying 295 groups to perform this calculation. From a computation time point of view, this is very interesting. However, it might be difficult to handle other calculations, such as depletion.

4.3.2 `SYBILT:` path: comparison of the Eigenvalues

In this section, results are shown for the cross-sections generated in DRAGON with the method of collision probabilities and interface currents. In INSTANT, calculations were performed in P_3 with linear anisotropic scattering in both cases.

Supercells, homogeneous hexagons, 295 groups, SYB

Table 4.15(a) summarizes the results for every group-structure used in INSTANT, when the cross-sections were generated in DRAGON with 295 groups. The results from the single block path are shown next to it (table 4.15(b)).

Table 4.15 – Comparison of Eigenvalues between the supercell and single block paths, `SYBILT:` solver in DRAGON starting with 295 groups

(a) Supercells, case SYB HOM295

Nb of groups in INSTANT	SC SYB HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24918	-36
26	1,24932	-25
23	1,24932	-25
12	1,25028	52
10	1,25053	72
9	1,25029	53
6	1,25069	85
4	1,24582	-305
2	1,24884	-63

(b) Single block, case SYB HOM295

Nb of groups in INSTANT	SB SYB HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,25024	49
26	1,25031	54
23	1,24994	25
12	1,24993	24
10	1,25011	38
9	1,24691	-218
6	1,24776	-150
4	1,20897	-3254
2	1,21184	-3024

The Eigenvalues show a good agreement with MCNP until 4 groups are used in INSTANT. The results are within ± 40 pcm with 295 to 23 groups, and with 12, 10, 9 or 6 groups they are still at less than 90 pcm from the MCNP reference. This is curious, because 9 or 6 groups seemed to be not enough when the cross-sections were computed in DRAGON using the MOC solver. It may be a coincidence resulting from cancellation of errors. It is also noticed here that the supercell path is providing a significantly better agreement with MCNP when a small number of groups is used in INSTANT (2 or 4 groups). The particularly good result with the supercell path and 2 groups may come from cancellation of errors since the lattice calculations with `SYBILT:` were not well comparing to the corresponding MCNP reference.

Those results with cross-sections generated by `SYBILT:` are interesting because recall that the agreement between MCNP and DRAGON was rather bad at the lattice level. It means therefore that there is some cancellation of errors leading to such a good agreement. One may also infer that the flux calculated at the lattice level might be quite good in the middle of the domain, but worse at the periphery. It would explain that the cross-sections coming from the supercells are more or less accurate enough to give a good Eigenvalue in INSTANT with 26 groups. This agreement still has to be also evaluated using the fission rate to make

complete statements.

Supercells, homogeneous hexagons, 26 groups, SYB

Table 4.16(a) summarizes the results for every group-structure with cross-sections generated in DRAGON starting with 26 groups in the supercell path. The corresponding K_{eff} values obtained with the single block path are recalled next to it in table 4.16(b).

Table 4.16 – Comparison of the Eigenvalues between supercells and single block paths, `SYBILT:` solver in DRAGON with 26 groups

(a) Supercells, case SYB HOM26

Nb of groups in INSTANT	SC SYB HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,24951	-10
10	1,25051	70
6	1,25080	94
4	1,24631	-266
2	1,24893	-56

(b) Single block, case SYB HOM26

Nb of groups in INSTANT	SB SYB HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25074	89
10	1,25099	109
6	1,24765	-158
4	1,21241	-2978
2	1,21243	-2977

Again, the agreement is good with 26 groups, and becomes poor with 4 or 2 groups. The Eigenvalues are comparable both to the corresponding results of the paths using MOC and supercells, but also to the path using supercells, `SYBILT:` and 295 groups in DRAGON. It would appear that the `SYBILT:` solver may be useful even if the agreement with MCNP was poor at the lattice level. The supercell path is also leading to clearly better results with 2 or 4 groups compared to the single block path.

In order to make more complete remarks on the different ways of generating cross-sections using supercell, we will now look at the fission maps and compare it to the MCNP reference as we did for the calculations of the first path.

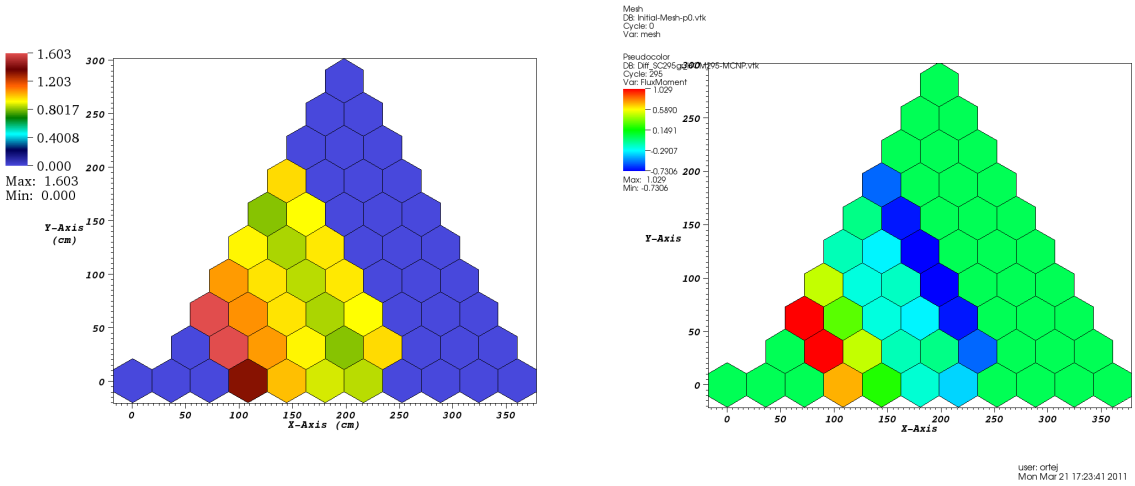
4.3.3 Fission rate maps

General overview of the fission maps for the second path

Figure 4.11 presents the block averaged fission rates for the case SC MOC HOM295, 295 groups in INSTANT, and the differences observed with the MCNP reference.

The fission rate maps look quite the same as for the first path. One can observe the two thermal peaks in the fuel regions adjacent to the reflector, with a higher peak in the inner

part. The highest errors are located in the peripheral fuel blocks with an over-prediction of the fission rates at the inner interface with the reflector and an under-prediction at the outer interface. On the contrary, the middle of the annular fuel region agrees well with the reference.



(a) Block-averaged fission rates (b) Percent error on block-averaged fission rate

Figure 4.11 – Supercell path - SC MOC HOM295, 295 groups in INSTANT - Block-averaged fission rates and associated error compared to the MCNP reference calculation

Extensive study of the fission maps

The following tables summarize the results with maximal positive errors (Max), maximal negative errors (Min), average of the errors (Av. μ), its standard deviation of errors (Std σ) and the absolute deviation of errors (AbsD.).

Supercells, MOC, HOM295: Table 4.17(a) presents the results obtained when the supercell calculations were performed in 295 groups in DRAGON and using the method of characteristics. Intuitively, this calculation should provide the best results because the cross-sections have to some extent been generated with a spectrum more representative of the real environment in the core. This is, after all, the idea of using supercells. The MOC solver should also be more accurate than `SYBILT:`. It was also demonstrated on the single block that the pre-calculation needed to perform the supercell calculation is accurately carrying the double-heterogeneity effect of the TRISO, so that information is not lost by doing this.

Table 4.17 – Fission map study, single block versus supercell paths

(a) Supercells, case MOC HOM295

Nb of groups in INSTANT	SC MOC HOM295 P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD.
295	1,03	-0,73	-0,08	0,51	0,44
26	0,89	-0,75	-0,07	0,49	0,40
23	1,12	-0,61	-0,09	0,49	0,43
12	0,64	-0,69	-0,05	0,41	0,34
10	0,67	-0,64	-0,05	0,39	0,34
9	0,48	-0,39	-0,01	0,22	0,18
6	0,51	-0,58	-0,02	0,31	0,26
4	5,50	-5,89	0,53	3,74	3,35
2	6,86	-5,89	0,54	4,37	3,94

(b) Single block, case MOC HOM295

Nb of groups in INST.	SB MOC HOM295 P_3 scat 1			
	Max	Min	Av. μ	St σ
295	1,86	-0,99	-0,14	0,82
26	1,82	-1,01	-0,14	0,86
23	1,85	-1,05	-0,14	0,91
12	1,29	-0,93	-0,10	0,75
10	1,33	-0,97	-0,10	0,78
9	0,35	-0,51	-0,007	0,28
6	0,78	-0,71	-0,06	0,49
4	11,2	-14,5	0,99	9,46
2	10,8	-14,4	1,01	9,31

Next to table 4.17(a), table 4.17(b) recalls the maxima, minima, averages and standard deviation obtained with the single block path. The absolute deviation can be found in table 4.8 (page 107). To be consistent in the comparison, the path where the same set of initial homogenized cross-sections were used in DRAGON was selected for the single block calculation, but it does not differ significantly from the results where the DRAGON calculation was directly performed with the TRISO particles.

An improvement is clearly observed with the supercells. For example, in 26 groups, the maximal error has been divided by 2 using supercells, and the minimal error is reduced by 25 %. The averaged error is divided by two and the standard deviation is also reduced by 40 %. The supercell path appears to be better for all group structures except the 9-group one. This may further the argument that the good results obtained with the single-block calculations collapsed to 9-groups was due to fortuitous cancellation of errors.

The supercell path leads to fission rates with errors located in general within ± 1 % compared to MCNP. This is considered a very good agreement. The results are good with 295 groups to 6 groups in INSTANT. Again, there is curiously very good agreement on the fission rates with the 9-group and 6-group calculations while their Eigenvalues were much farther away from the MCNP reference than the results with finer group-structures.

Supercells, MOC, HOM26: Table 4.18(a) presents the results obtained from starting the supercell calculations in DRAGON with 26 groups. The corresponding results obtained by the single block path are recalled in table 4.18(b). Both DRAGON calculations were performed with the method of characteristics.

Table 4.18 – Supercell fission rate study and comparison with the single block path, DRAGON calculation done in 26 groups

(a) Supercells, case MOC HOM26

Nb of groups in INSTANT	SC MOC HOM26				
	P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD.
26	1,18	-0,90	-0,10	0,62	0,53
10	0,88	-0,85	-0,07	0,54	0,45
6	0,62	-0,63	-0,02	0,39	0,35
4	4,53	-5,08	0,46	3,20	2,90
2	6,58	-5,70	0,53	4,2	3,81

(b) Single block, case MOC HOM26

Nb of groups in INST.	SB MOC HOM26			
	P_3 scat 1			
	Max	Min	Av. μ	St σ
26	2,05	-1,03	-0,16	0,87
10	1,47	-0,91	-0,11	0,71
6	1,13	-0,69	-0,09	0,55
4	9,90	-12,7	-0,85	8,27
2	10,6	-14,2	-0,98	9,19

Here, an improvement is again observed when supercells are used. When the INSTANT calculation is performed in 26 groups, the maximal error is divided by 1,7 and the minimal error is lowered slightly. The decrease is more impressive in the 10-group calculation, but this might be a coincidence. Also note that compared to the supercell path performed in 295 groups in DRAGON, the loss of accuracy is not very important. For example, the maximal error goes from 0,89 to 1,18. Given the calculation times, it might be interesting to accept this compromise because performing a supercell calculation in DRAGON with 295 groups is time consuming and requires a lot of data exchange.

Supercell, SYB, HOM295: These calculations were repeated using cross-sections calculated from `SYBILT:`. Table 4.19(a) describes the results obtained when the supercell calculations were done in 295 groups. The corresponding results from the single block path (table 4.11) are partially recalled here in table 4.19(b).

This shows that the group structures from 295 groups to 6 groups give almost the same results. Here the best group-structure regarding the fission map seems to be the 6-group structure. Again, this was not expected because the K_{eff} is not the best one for this group-structure.

Here the advantage of the supercells is less evident. For example, with 26 groups in INSTANT, the over-prediction is reduced by 79 %, but at the same time, the under-prediction is increased to -1,53 % compared to -0,99 %. The standard and absolute deviations are about the same in both paths. The average of the error is a bit better for the supercell path, which may indicate that the fission rates are better centred on the reference's values, but with the same deviations. No clear gain is therefore obtained by the supercells here.

Recalling that the agreement was quite poor with MCNP at the supercell level using

Table 4.19 – Supercells and single block paths comparison of the fission rate, `SYBILT:` solver in DRAGON

(a) Supercells, case SYB HOM295

Nb of groups in INSTANT	SB SYB HOM295				
	P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD.
295	1,00	-1,49	-0,046	0,81	0,68
26	1,01	-1,53	-0,039	0,83	0,69
23	0,89	-1,29	-0,053	0,72	0,59
12	0,96	-1,44	-0,01	0,77	0,64
10	0,92	-1,38	-0,014	0,74	0,61
9	0,79	-1,20	0,039	0,69	0,63
6	0,78	-1,11	0,015	0,62	0,53
4	4,38	-8,29	0,69	3,40	2,62
2	4,75	-8,45	0,70	3,45	2,66

(b) Single block, case SYB HOM295

Nb of groups in INST.	SB SYB HOM295			
	P_3 scat 1			
	Max	Min	Av. μ	St σ
295	1,83	-0,97	-0,14	0,81
26	1,79	-0,99	-0,13	0,84
23	1,83	-1,04	-0,14	0,89
12	1,26	-0,90	-0,10	0,72
10	1,30	-0,95	-0,10	0,76
9	0,36	-0,55	-0,009	0,29
6	0,75	-0,69	-0,06	0,47
4	11,2	-14,5	0,99	9,46
2	10,8	-14,4	1,01	9,32

`SYBILT:`, perhaps the flux calculation is not accurate enough at this stage to produce cross-sections which would increase the accuracy of the full-core calculation as observed with the MOC path. Compared to this path, it is observed that the loss of accuracy is essentially located on the maximal negative error.

Supercell, SYB, HOM26: Table 4.20(a) shows the results when the DRAGON calculation was performed in 26 groups using `SYBILT:`. Table 4.20(b) recalls some of the corresponding results with the single block path.

Table 4.20 – Supercells and single block paths comparison on fission rates with `SYBILT:` and DRAGON calculation starting with 26 groups

(a) Supercells, case SYB HOM26

Nb of groups in INSTANT	SB SYB HOM26				
	P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD.
26	2,04	-1,02	-0,065	0,88	0,71
10	1,46	-0,90	-0,036	0,87	0,70
6	1,12	-0,68	0,011	0,74	0,61
4	9,90	-12,7	0,59	2,95	2,24
2	10,6	-14,3	0,67	3,30	2,55

(b) Single block, case SYB HOM26

Nb of groups in INST.	SB SYB HOM26			
	P_3 scat 1			
	Max	Min	Av. μ	St σ
26	2,04	-1,02	-0,15	0,87
10	1,46	-0,90	-0,11	0,70
6	1,12	-0,68	-0,08	0,54
4	9,90	-12,7	-0,85	8,27
2	10,6	-14,3	-0,98	9,19

Again, the results with supercells are good with 26, 10 and 6 groups with a best agreement

with 6 groups. In 4 and 2 groups, the fission rates are poor. One does not see any difference on the width of the error committed in the two paths compared to MCNP.

Compared to the equivalent calculations where 295 groups were kept in DRAGON, the loss of accuracy is not totally clear. The maximal error is multiplied by 2 when 26 groups are used in DRAGON instead of 295, but at the same time, the minimal error goes from -1,53 % in 295 groups to -1,02 %. One can anyway note that the average error is lower when DRAGON was using 295 groups.

4.3.4 Results with the SPH technique

In the previous core calculations, only a direct homogenisation was used in DRAGON to produce the homogeneous sets of cross-sections for INSTANT. As far as a single block is considered, the whole lattice is homogenized making the SPH factors all equal to one. This is not the case for supercells were only the central block is homogenised. This section describes the results obtained in INSTANT using the sets of cross-sections corrected by the SPH technique described in sections 2.11.2 and 3.6.2. Results involving MOC calculations in DRAGON are first presented, and then the results with the `SYBILT:` solver. In each table, we recall side by side on the right the results obtained from the path without the SPH technique in DRAGON but the same solver, so that the effect of SPH can be clearly noticed.

Comparison of the Eigenvalues

Table 4.21 – Comparison of the Eigenvalues between supercells paths with and without SPH technique, MOC solver in DRAGON with 295 groups

(a) Supercells with SPH, case MOC HOM295

Nb of groups in INSTANT	SC MOC HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24961	-2
26	1,24981	14
23	1,24977	11
12	1,25054	73
10	1,25072	87
9	1,24836	-101
6	1,25168	164
4	1,25680	574
2	1,25805	674

(b) Supercells without SPH, case MOC HOM295

Nb of groups in INSTANT	SC MOC HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24953	-8
26	1,24984	17
23	1,24980	14
12	1,25099	109
10	1,25126	130
9	1,25198	188
6	1,25231	214
4	1,25939	781
2	1,26236	1019

Table 4.22 – Comparison of the Eigenvalues between supercells paths with an without SPH technique, MOC solver in DRAGON with 26 groups

(a) Supercells with SPH, case MOC HOM26

Nb of groups in INSTANT	SC MOC HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25009	37
10	1,25074	89
6	1,25208	196
4	1,25642	543
2	1,25837	699

(b) Supercells without SPH, case MOC HOM26

Nb of groups in INSTANT	SC MOC HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,25003	32
10	1,25118	124
6	1,25273	248
4	1,25832	695
2	1,26230	1014

It is observed that the use of SPH techniques is almost always leading to an improvement of the Eigenvalue when it is compared to the case with the same number of groups but no super homogenisation. When no improvement is observed (26 groups in tables 4.22), the Eigenvalue is very similar between the two cases.

The SPH technique seems to be especially efficient with a low number of groups: for calculations with 2 to 9 groups, the improvement is particularly interesting. A calculation done with SPH and 6 groups leads to a better Eigenvalue than a calculation with 9 groups and no SPH (see tables 4.21).

Table 4.23 – Comparison of Eigenvalues between the supercell paths with and without SPH technique, `SYBILT:` solver in DRAGON starting with 295 groups

(a) Supercells with SPH, case SYB HOM295

Nb of groups in INSTANT	SC SYB HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24916	-38
26	1,24924	-31
23	1,24910	-42
12	1,24991	22
10	1,25011	38
9	1,24991	22
6	1,25012	39
4	1,24530	-347
2	1,24731	-186

(b) Supercells without SPH, case SYB HOM295

Nb of groups in INSTANT	SC SYB HOM295 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
295	1,24918	-36
26	1,24932	-25
23	1,24932	-25
12	1,25028	52
10	1,25053	72
9	1,25029	53
6	1,25069	85
4	1,24582	-305
2	1,24884	-63

Table 4.24 – Comparison of the Eigenvalues between supercells paths with or without SPH technique, `SYBILT:` solver in DRAGON with 26 groups

(a) Supercells with SPH, case SYB HOM26

Nb of groups in INSTANT	SC SYB HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,24943	-16
10	1,25004	33
6	1,25021	46
4	1,24574	-311
2	1,24752	-169

(b) Supercells without SPH, case SYB HOM26

Nb of groups in INSTANT	SC SYB HOM26 P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)
26	1,24951	-10
10	1,25051	70
6	1,25080	94
4	1,24631	-266
2	1,24893	-56

The use of the `SYBILT:` solver in DRAGON leads to similar results in some ways. In the case where the DRAGON calculation is done starting with 295 groups (table 4.23), the core calculations with 295 groups and 26 groups show no improvement when SPH is used at the lattice level, but the Eigenvalues remain similar between both cases. It is improved with 23 to 6 groups, as in the MOC cases. However, SPH causes to deteriorate more significantly the results with 4 and 2 groups. This deterioration may come from the inaccuracy of the `SYBILT:` tracking on supercells, so that the conservation of the reaction rates ensured by the SPH technique lacks precision to be efficient: if the heterogeneous reaction rates are already rather wrong at the lattice level, then their conservation does not make sense any more and may even more deteriorate the following core calculation.

The same analyse can be done on the case using cross-section condensed at 26 groups before the supercell calculation (table 4.24).

It is concluded from this Eigenvalue study that the SPH technique is in general very slightly improving the error made on the Eigenvalue. This is especially true for calculations with few groups and when MOC was used in DRAGON. This may be due to more precise reference reaction rates computed by DRAGON before doing the super-homogenisation. In some cases, it may be interesting to use SPH instead of increasing the number of groups in the core calculation.

Fission rate maps

The same presentation is used here to compare the fission rate maps of supercell paths with or without SPH technique: the table on the left provides the results of the fission rate map study with cross-sections corrected by SPH in DRAGON, while the table on the right

recalls the results of the same case without the use of SPH.

Table 4.25 – Comparison of the fission rates, supercell paths with or without SPH, MOC solver in DRAGON with 295 groups

(a) Supercells with SPH, case MOC HOM295

Nb of groups in INSTANT	SC MOC HOM295 P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD.
295	3,12	-2,57	0,17	1,82	1,62
26	3,28	-2,80	0,20	2,05	1,90
23	3,02	-2,65	0,17	1,76	1,56
12	4,35	-4,04	0,33	3,03	2,93
10	4,39	-4,11	0,33	3,04	2,93
9	6,50	-16,1	0,05	6,19	4,66
6	5,08	-5,14	0,41	3,30	3,12
4	7,91	-10,4	0,93	4,85	3,55
2	10,3	-13,2	1,10	5,92	4,56

(b) Supercells without SPH, case MOC HOM295

Nb of groups in INST.	SC MOC HOM295 P_3 scat 1			
	Max	Min	Av. μ	Std σ
295	1,03	-0,73	-0,08	0,51
26	0,89	-0,75	-0,07	0,49
23	1,12	-0,61	-0,09	0,49
12	0,64	-0,69	-0,05	0,41
10	0,67	-0,64	-0,05	0,39
9	0,48	-0,39	-0,01	0,22
6	0,51	-0,58	-0,02	0,31
4	5,50	-5,89	0,53	3,74
2	6,86	-5,89	0,54	4,37

Table 4.26 – Comparison of the fission rates, supercell path with and without SPH, MOC solver in DRAGON with 26 groups

(a) Supercells with SPH, case MOC HOM26

Nb of groups in INSTANT	SC MOC HOM26 P_3 scat 1				
	Max	Min	Av. μ	St σ	AbsD.
26	3,07	-2,46	0,15	1,83	1,63
10	4,17	-3,66	0,29	2,95	2,84
6	5,19	-4,73	0,39	3,26	3,07
4	7,47	-9,47	0,83	4,33	3,09
2	9,85	-12,6	1,05	5,65	4,31

(b) Supercells without SPH, case MOC HOM26

Nb of groups in INST.	SC MOC HOM26 P_3 scat 1			
	Max	Min	Av. μ	St σ
26	1,18	-0,90	-0,10	0,62
10	0,88	-0,85	-0,07	0,54
6	0,62	-0,63	-0,02	0,39
4	4,53	-5,08	0,46	3,20
2	6,58	-5,70	0,53	4,2

Table 4.27 – Comparison of the fission rates, supercell path with or without SPH, `SYBILT:` solver in DRAGON with 295 groups

(a) Supercells with SPH, case SYB HOM295

Nb of groups in INSTANT	SB SYB HOM295 P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD.
295	5,18	-4,09	0,34	3,41	3,23
26	5,48	-4,61	0,38	3,63	3,47
23	5,54	-4,57	0,38	3,61	3,44
12	6,19	-5,71	0,47	4,12	3,94
10	6,20	-5,74	0,47	4,11	3,93
9	6,79	-7,06	0,57	4,51	4,29
6	7,06	-6,89	0,57	4,55	4,30
4	10,2	-13,3	1,12	6,53	5,59
2	12,1	-15,2	1,26	7,61	6,65

(b) Supercells without SPH, case SYB HOM295

Nb of groups in INST.	SB SYB HOM295 P_3 scat 1			
	Max	Min	Av. μ	Std σ
295	1,00	-1,49	-0,046	0,81
26	1,01	-1,53	-0,039	0,83
23	0,89	-1,29	-0,053	0,72
12	0,96	-1,44	-0,01	0,77
10	0,92	-1,38	-0,014	0,74
9	0,79	-1,20	0,039	0,69
6	0,78	-1,11	0,015	0,62
4	4,38	-8,29	0,69	3,40
2	4,75	-8,45	0,70	3,45

Table 4.28 – Comparison of the fission rates, supercell paths with or without SPH, `SYBILT:` solver in DRAGON with 26 groups

(a) Supercells with SPH, case SYB HOM26

Nb of groups in INSTANT	SB SYB HOM26 P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD.
26	5,27	-4,00	0,34	3,48	3,30
10	5,93	-5,12	0,42	3,98	3,82
6	7,07	-6,52	0,54	4,84	4,23
4	9,97	-12,4	1,04	6,21	5,37
2	11,5	-14,8	1,21	7,22	6,30

(b) Supercells without SPH, case SYB HOM26

Nb of groups in INST.	SB SYB HOM26 P_3 scat 1			
	Max	Min	Av. μ	Std σ
26	2,04	-1,02	-0,065	0,88
10	1,46	-0,90	-0,036	0,87
6	1,12	-0,68	0,011	0,74
4	9,90	-12,7	0,59	2,95
2	10,6	-14,3	0,67	3,30

What ever the tracking method used in DRAGON (MOC or `SYBILT:`), it is observed that the SPH method is always increasing the errors in a very significant way. The different parameters are multiplied by factors between 3 and more than 10. The fission maps are deteriorated by the use of SPH in the lattice calculations of the supercells compared to the direct homogenisation. This result, opposed to the trend observed on the Eigenvalues, was unexpected.

As a first explanation, it is pointed out that the homogeneous calculation done with MOC or PIJ methods in DRAGON during the SPH iterations may give a poor representation of the homogeneous solution that would come out of an INSTANT homogeneous lattice calculation. Therefore, doing the SPH iterations with INSTANT may be compulsory to clearly evaluate the efficiency of the super-homogenisation technique.

One could also suggest the following explanation: the SPH homogenisation ensures the conservation of the reaction rates computed at the lattice level between the heterogeneous and homogeneous geometries. But it may happen that the reaction rates of the lattice heterogeneous calculation, that serve as a reference, do not correspond to the true reaction rates of each block seeing the reflector at the core level. Therefore, the better agreement of the Eigenvalues may be due to compensation of errors.

If true, this explanation may also suggest that the use of supercells may be more successful if a first study is done to compare the spectra obtained by MCNP at the core level and by DRAGON at the lattice level. It would then ensure that a supercell modelled in DRAGON is really representative of the block that it intends to represent. Once this analysis has been completed, a super-homogenisation may make much more sense because it would conserve the right reaction rates regarding each homogeneous block of the core calculation.

4.3.5 Conclusion of the 2nd path: supercells

Contrary to the single block path, the interest of supercells differs depending on the solver which was employed in DRAGON to produce the homogenized cross-sections. The MOC solver gives clearly better results on the shape of the fission rate compared to the single block model. This is especially true when the supercells were calculated in DRAGON with 295 groups. A reduction of about 62 % of the interval width of the errors compared to MCNP is observed in this case. Most of the improvement is seen in the over-predictions of the fission rates in the fuel blocks located next to the inner reflector, which is almost divided by a factor 2.

When DRAGON is using 26 groups and the MOC, the errors are increased but remain acceptable and better than the corresponding calculations with the single block model. Thus, a consistent trend of improvement is seen due to the use of supercells. The results of the

supercell path with 26 groups in DRAGON are even better than the results of the single block path with 295 groups in DRAGON.

The prediction of the Eigenvalue is also better in the supercell path with MOC when the same number of groups was used in the single block model in DRAGON. The advantage is here more clear for keeping 295 groups: we obtain a deviation to MCNP equal to only -8 pcm, which is within the statistical uncertainty of the MCNP calculations.

On the contrary, the use of `SYBILT:` to perform the calculations in DRAGON does not lead to a clear improvement of the INSTANT results. The fission rate map is not centred around the same value so that the maximal error is decreased but the minimal error is increased. The K_{eff} values are also farther from the reference.

Regarding the SPH technique, it is observed that it slightly improves the Eigenvalues with MOC, especially when a small number of groups is kept at the core level. However, it deteriorates by a factor 3 to 10 and more the agreement of the fission rate maps with MCNP. This shows that the SPH homogenisation is not reliable in this case. Another equivalence such as MOC-SP_3 might give better results but it was not tested in a core calculation.

4.4 Calculation durations

An INSTANT calculation with the linear anisotropic scattering matrix in 295 groups takes 123 seconds in P_1 and 635 seconds in P_3. With only 26 groups, it takes 5 seconds in P_1 and 37 seconds in P_3. Those values are rather independent of the way the cross-sections are prepared because the number of iterations is generally the same whatever the path chosen for cross-section generation.

In DRAGON, the condensation has an influence of about 1 or 2 minutes, depending on the amount of data the code has to write in ASCII files. It is clearly not significant compared to the rest of the calculation. Therefore, only the calculation durations are given for cases with a condensation to 26 groups at the end. Table 4.29 concerns DRAGON calculations made on a single block whereas table 4.30 concerns supercell calculations in DRAGON.

Table 4.29 – Single block: Summary of calculation durations in DRAGON

Calculation durations in DRAGON, Single block			
	MOC DH295	MOC HOM295	MOC HOM26
Duration	46 min	8 min	31s
	SYB DH295	SYB HOM295	SYB HOM26
Duration	6 min	19s	2s

Table 4.30 – Supercells: Summary of calculation durations in DRAGON

Calculation durations in DRAGON, Supercells			
	MOC DH295	MOC HOM295	MOC HOM26
Duration	–	82h10min	2h39min
	SYB DH295	SYB HOM295	SYB HOM26
Duration	–	7h58min	7min

As can be seen from those tables, the calculations with the method of characteristics are very time-consuming, especially for the supercells when using 295 groups. Not surprising is the fact that this method is also the most accurate. The `SYBILT:` solver is fast but does not give the same accuracy and cannot serve as a reference calculation. The use of SPH do not introduce a big difference on the order of magnitude of the calculation times.

It can be seen that the use of supercells is handicapped by the time constraint. The parallelization of the MOC solver may be of great value if the supercells are determined to be required for calculations in this core. An integrated environment for the lattice and core calculations may also help a lot.

4.5 Conclusion on the cross-sections generation

Considering only a steady-state calculation with fresh fuel, there is a clear advantage to generating microscopic homogenized cross-sections from supercell models in DRAGON if the MOC solver is used. However, regarding the calculation durations and the complexity of the use of supercells, it is not clear that the gain of accuracy is significant enough to justify their use. The single block model already provides a good solution on the condition that enough groups are used in INSTANT: typically 12 or more. One should remember here that the interest of supercells may be higher when burnable poisons and control rods are added in the core models.

A partial explanation for the fact that supercells do not make a significant difference compared to the single block path can be found by the study of the neutron energy spectra in both cases. The word spectrum describes the flux as a function of the energy, normalised such as the integral of this curve is equal to 1. Because a multi-group approximation was used, the flux in 295 groups is first divided by a lethargy interval:

$$\frac{\phi(E_i)}{\log_{10}(E_{i+1}) - \log_{10}(E_i)},\ i \in [\![1, 295]\!] \tag{4.3}$$

This term is plotted as a function of the base 10 logarithm of the energy (in eV):

$\log_{10}(E_{\text{central},i})$. $E_{\text{central},i}$ is calculated using the upper and lower boundaries of the group:

$$E_{\text{central},i} = 10^{\left(\frac{\log_{10}(E_{i+1})+\log_{10}(E_i)}{2}\right)} \tag{4.4}$$

Note that what is called "lethargy" in the figures is not equal to the usual definition of the lethargy. It has been used because the fluxes are plotted on a logarithmic scale. It is known that the wider is a group, the more important will be the flux, so that to be able to compare the level of the flux in each group, it is necessary to divide the fluxes by the logarithmic widths of their groups.

Before plotting it, a normalisation is performed so that the integral of the function on a logarithmic scale is equal to one. Therefore the fluxes are normalized by:

$$\begin{aligned}\tilde{\phi}(E_i) &= \frac{1}{\sum_{i=1}^{295} \frac{\phi(E_i)}{\log_{10}(E_{i+1})-\log_{10}(E_i)} \cdot (\log_{10}(E_{i+1})-\log_{10}(E_i))} \times \frac{\phi(E_i)}{\log_{10}(E_{i+1})-\log_{10}(E_i)} \\ &= \frac{1}{\sum_{i=1}^{295} \phi(E_i)} \times \frac{\phi(E_i)}{\log_{10}(E_{i+1})-\log_{10}(E_i)}\end{aligned} \tag{4.5}$$

This normalisation enables comparison of several spectra together so that neutron thermalization can be observed in each model. All the spectra have an integral equal to 1 on a logarithmic scale.

Figure 4.12 contains the spectra for the single block (in blue) and the supercell path (in red).

This shows that the spectra of the single block and of the supercell type 18 look almost the same. The supercell number 18 is made of a block surrounded by 3 reflector blocks and 3 fuel blocks. This result was unexpected. It shows that on average, the flux looks the same over the block. The thermal peaks that appears on the right of the supercell next to the reflector are compensated for by the higher fast neutron flux coming from the left side, so that on average the spectrum of the single block is approximately the same. Therefore, the homogenization of the cross-sections will not differ much between the two models, but this may become different after some depletion steps.

Notice that for the two other types of supercells which are surrounded by 2 blocks of reflector (cell 19) or 1 block of reflector (cell 13), the spectra significantly differ. For them, there is a clear thermal peak which is higher when there are two blocks of reflector compared to 1 block. The thermal peaks are also higher than the one observed for the single block or the supercell surrounded by 3 blocks of reflector.

As seen in the detailed fission rate map, the blocks next to the reflectors show a high

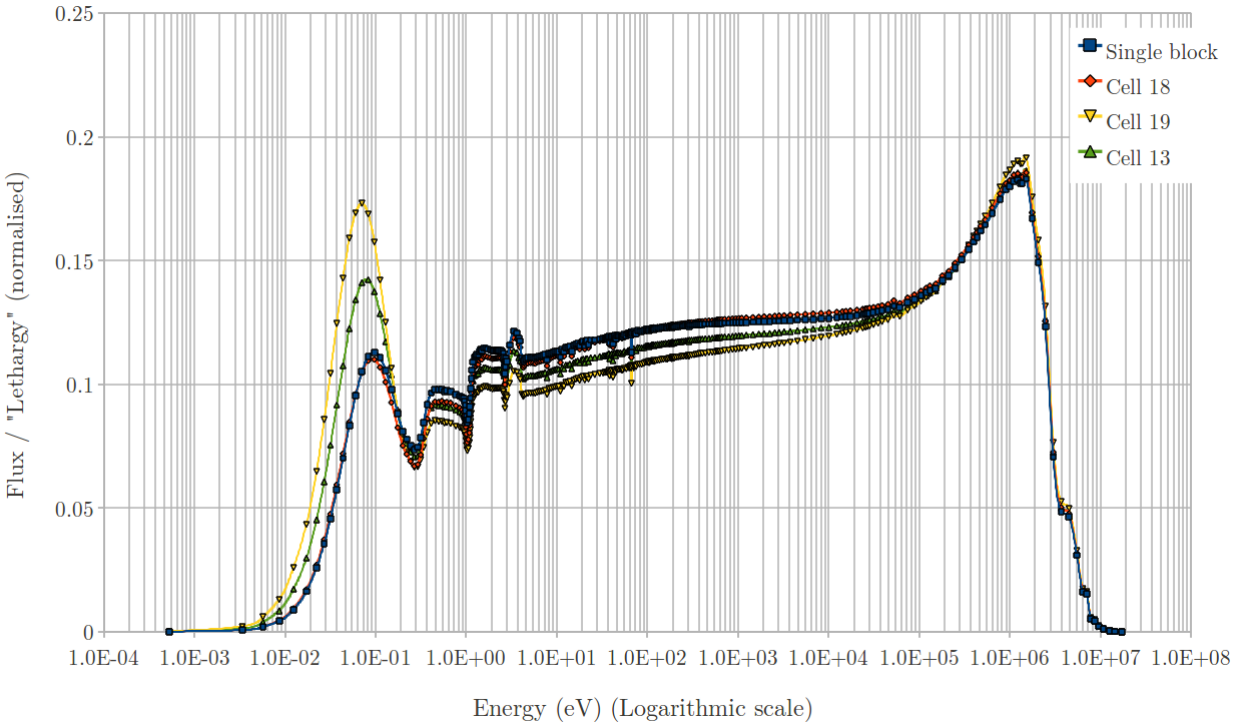


Figure 4.12 – Comparison of the spectra averaged over the block of interest between the single block and the 3 types of supercell (Cells 18, 19 and 13). 295 groups, homogeneous cells, `SYBILT:`. Cell 18 is surrounded by 3 reflector blocks, cell 19 by 2 reflector blocks and cell 13 by 1 reflector block.

variation of their fission rates in their halves in contact with the reflector blocks. This indicates that in these regions, the spectrum may change a great deal. This is also confirmed by the spectral plots that will be shown for the depletion study (see section 4.7 and figure 4.20 later in the text). Considering that the two parts seems to behave very differently, using an averaged of the cross-sections over the entire block of interest may not be acceptable because this information on the spectral variation is lost.

Therefore, one recommendation would be to homogenize the two parts separately using, for example, 6 triangular sectors per block and annular regions in the block. It would then be possible to employ different sets of cross-sections for a block located near the reflector in INSTANT. This way of generating cross-sections for the core calculation may lead to more accurate results than those presented here because it would then fully utilize the information provided by a supercell model. This may be particularly true when burnable poisons or control rods are added. This additional level of detail was beyond the scope of this work.

The use of SPH introduces improvement on the Eigenvalues but causes to deteriorate the fission rate maps. Partial explanation was given, such as the fact that the supercells may not well represent some of the blocks seeing the reflector. Thus, conservation of wrong reaction rates is leading to a local deterioration or improvements, while on the whole, some integrated parameters such as the Eigenvalue may be similar or better than the one obtained with a direct homogenisation. More studies should be performed to assess this hypothesis. Another explanation is that the MOC-MOC equivalence is not adapted to the following INSTANT calculation because the two methods are too much different. It is possible that this equivalence introduces more errors in the reaction rates than corrections. Finally, the MOC-MOC equivalence does not always converge with few groups during the SPH algorithm, as it is established in the following section (4.6). For all those reasons, it is believed that the improvement seen on the Eigenvalues with SPH results from compensation of errors. Finer analysis should be pursued with for example MOC-INSTANT SPH equivalences.

4.6 Validation of some SPH calculations in DRAGON

4.6.1 Method of investigation and results

In order to investigate the efficiency of the SPH technique, tests were done in DRAGON and DONJON on a supercell with 3 reflector blocks to validate the SPH routines. The corresponding homogeneous geometry was designed to perform the SPH equivalence calculations when needed. Notice here that the homogeneous calculation were performed within DONJON using the modules BIVAC and TRIVAC. INSTANT was never used for this validation which was pursued after our internship in Idaho Falls. However, it is assumed that the

TRIVAC module of DONJON with SP_N is reasonably representative of the P_N methods of INSTANT. Details on the implementation of SP_N methods can be found in reference [32].

The tests consist in the following steps:

- Performing the transport calculation with MOC on the heterogeneous geometry. Saving the flux object for further use and store the Eigenvalue and homogenised absorption rates provided by the heterogeneous solution. The reaction rates are condensed to 2 groups to make comparisons easier with homogeneous cases and they are taken as the reference.
- Creating a reactor database without SPH equivalence. The cross-sections are homogenized over each block of the supercell geometry and condensed to 26, 6 or 2 groups.
- Creating a reactor database with SPH equivalence. The SPH equivalence procedure was tested with different methods for the homogeneous calculation: MOC, diffusion, SP_1 or SP_3 with isotropic scattering and the keyword DIFF, or without this keyword. DIFF enables the calculation of the total cross-section from the homogenised diffusion coefficient instead of recovering it from the homogenisation.
- Performing two homogeneous calculations for each resolution method: one with SPH-corrected cross-sections and one without SPH correction. A comparison of the absorption rates and Eigenvalues to the reference enables one to verify that the SPH procedure is correctly working in DRAGON. As a matter of fact the SPH correction should lead to about 0% of difference with the reference, while a direct homogenisation does not automatically preserve the reaction rates.

The results are presented in Tables 4.31 and 4.32. The deviation of the Eigenvalue to its reference is of course a necessary condition to assess that the equivalence procedure was correctly performed in DRAGON, but the analysis of the deviations on the absorption rates is also essential to prove the convergence of the super-homogenization.

Table 4.31 – Validation of the SPH technique in DRAGON for 2 and 6 groups

(a) Condensation to 2 energy groups

Method	Absorption rates deviation		K_{eff} deviation (pcm)
	Max (%)	Min (%)	
Diffusion w/o SPH	37,4	-4,2	226
Diffusion with SPH	0,9	-1,1	-12
SP1 DIFF w/o SPH	37,9	-4,2	230
SP1 DIFF with SPH	0,9	-1,1	-11
SP3 DIFF w/o SPH	37,4	-4,1	216
SP3 DIFF with SPH	0,9	-1,1	-11
MOC w/o SPH	4,8	-12,1	548
MOC with SPH	1,3	-14,6	39
SP1 no DIFF w/o SPH	5,3	-15,5	658
SP1 no DIFF with SPH	2,9	-55,6	226
SP3 no DIFF w/o SPH	5,6	-15,5	639
SP3 no DIFF with SPH	15,4	-49,5	369

(b) Condensation to 6 energy groups

Method	Absorption rates deviation		K_{eff} deviation (pcm)
	Max (%)	Min (%)	
Diffusion w/o SPH	54,8	-6,8	413
Diffusion with SPH	5,4	-0,3	-7
SP1 DIFF w/o SPH	54,8	-6,8	412
SP1 DIFF with SPH	5,3	-0,3	-6
SP3 DIFF w/o SPH	56,3	-6,6	368
SP3 DIFF with SPH	5,6	-0,4	-6
MOC w/o SPH	7,9	-4,4	361
MOC with SPH	4,5	-4,5	-4
SP1 no DIFF w/o SPH	6,0	-8,4	417
SP1 no DIFF with SPH	2,5	-31,8	8
SP3 no DIFF w/o SPH	4,4	-8,3	374
SP3 no DIFF with SPH	830,7	-46,9	33199

Table 4.32 – Validation of the SPH technique in DRAGON for 26 groups

Method	Absorption rates deviation		K_{eff} deviation (pcm)
	Max (%)	Min (%)	
Diffusion w/o SPH	42,0	-6,4	317
Diffusion with SPH	0,4	-0,5	-2
SP1 DIFF w/o SPH	42,0	-6,4	317
SP1 DIFF with SPH	0,4	-0,5	-2
SP1 no DIFF w/o SPH	8,9	-21,4	339
SP1 no DIFF with SPH	1,3	-14,3	18
SP3 DIFF w/o SPH	40,7	-6,3	275
SP3 DIFF with SPH	0,4	-0,5	-2
SP3 no DIFF w/o SPH	4,6	-18,1	298
SP3 no DIFF with SPH	1,2	-14,4	17
MOC w/o SPH	10,0	-3,8	284
MOC with SPH	0,1	-0,1	-3

Table 4.31(a) shows that only diffusion and SP_1/SP_3 calculations with the DIFF keyword are relevant when a 2-group condensation is chosen. For those cases, the SPH technique is reducing the deviation to the heterogeneous reaction rates as expected. Diffusion and SP_1 with DIFF keyword present very similar results, which is expected as theoretically those two calculations are equivalent. This point is established in many textbooks and was already discussed in section 2.12.3.

On the contrary, results with MOC show only partial improvement on the Eigenvalue. Results for SP_1/SP_3 without DIFF keyword show almost no real improvement. This was expected because the SPH algorithm is diverging in DRAGON for MOC, SP_1/SP_3 without DIFF keyword when the reactor databases are generated. It explains why poor agreement is seen on those cases when the homogeneous corresponding calculations are performed.

Table 4.31(b) shows about the same trend with a 6-group condensation. SPH is only efficient with diffusion, SP_1 with DIFF keyword and SP_3 with DIFF keyword. No convergence is achieved in the generation of the SPH-corrected databases with MOC and SP_1/SP_3 without DIFF keyword. However, for MOC, the error is significantly decreasing, which indicates that the convergence is not far.

Finally, table 4.32 shows convergence for all types of homogeneous calculations except for SP_1/SP_3 without DIFF keyword. However, the lack of precision of SP_1/SP_3 without DIFF keyword was not detected by the code, which did not raise any error message after the SPH iterations.

4.6.2 Conclusion on the SPH validation

To conclude, this study shows that a higher number of groups in the condensation makes the convergence of the SPH algorithm easier in DRAGON. With 2 or 6 groups, only diffusion and SP_1 or SP_3 with isotropic scattering and DIFF keyword produce correct SPH-corrected databases. With 26 groups, the MOC equivalence shows convergence as well.

As our core calculations were based on transport-transport equivalences with MOC for each condensations, it is probable that the cross-sections passed to INSTANT were not corrected so as to preserve the heterogeneous reaction rates. This may explain why poor agreement was seen on the fission maps in section 4.3.4. However, this equivalence proved to be valid for 26 groups while the same deterioration of the fission rates was observed in INSTANT compared to the MCNP reference. Therefore, other causes have to be found to explain this deterioration.

It is finally noticed that currently some methods still lead to problems in the SPH algorithm. The SP_1 and SP_3 methods without the DIFF keyword are working in the code with a high number of groups but the reaction rates are not correctly preserved, even if the code runs to the end without raising any errors. With few groups, the code is diverging and the validation confirms this divergence.

This study seems to suggest that the keyword DIFF is more appropriate for SPH equivalences. This keyword makes DRAGON recalculate the total macroscopic cross-section using the diffusion coefficient rather than directly using the homogenised total cross-section. Further work should be pursued on this point.

4.7 Depletion study

As all of the machinery to deal with supercell creation had been automated, it was possible to rather quickly perform an initial study of the depletion methods which may be used in the future for this reactor. A more complete study should be performed to improve understanding of this since the gradients in the inner and outer ring of fuel introduce a very complicated flux shape and perhaps some difficult challenges for the depletion procedures.

4.7.1 Overview of the problems

In light water reactors, the depletion is generally performed in the lattice code. It consists of:

- performing a calculation over an assembly at different burnup steps and boron histories;
- creating a reactor database where the homogenized cross-sections tabulated as a function of the burn up are stored;

- passing these to the full-core calculation. The first full-core calculation creates a power map. A time interval is declared in the full-core solver to allow calculation of the burnup of each assembly knowing its power and its composition. Each assembly has therefore potentially a different burnup.
- The new macroscopic cross-sections for each assembly are recovered from the database depending on the particular burnup of each assembly. The full-core calculation can then be performed for burnup step 1. It provides a new power map for the core, and the process can be started again for next burnup iteration.

It works particularly well for light water reactors (at least in a first approximation) because the spectra do not differ between the assemblies. This spectrum is well described by a lattice calculation in the fundamental mode because of the small mean free paths. Notice that discretizing an assembly into 4 regions of depletion is usually sufficient to ensure a reasonable accuracy because the fluxes remain relatively flat over an assembly. However this procedure raises two questions when applied to the VHTR.

The first question can be formulated this way: given the fact that in this reactor, the flux is peaked in the peripheral assemblies, the concentrations in the blocks may change in a very different manner while burning. Therefore, is it still possible to consider averaged number densities over each one of the blocks at the core level, or should the method keep track of the detailed variation of the isotopic composition in the blocks to ensure the accuracy of the calculation? According to [33], the size of the depletion regions is "chosen so that within them, the group parameters at start of life are constant (possibly as the result of some homogenisation procedure) and *the flux shapes are reasonably flat*" ([33, p. 282]). Therefore, a finer discretization of the peripheral blocks may be needed for depletion in VHTRs. However, the introduction of burnable poisons and of burnt assemblies may also sufficiently flatten the peaks to enable one to consider homogeneous properties over any block in the core. In order to provide some insights, the influence of the flux peaking on the depletion is studied below using lattice calculations. A study over a more realistic core should be done to completely answer this question, but it is beyond the scope of this thesis.

The second question, perhaps more difficult, is to determine whether for a given burnup of a fuel block or of a fuel cell the depletion will be the same whatever the spectrum that has been imposed on the region so that it can reach this burnup. This is probably true for light water reactors *because the neutron energy spectrum is quite similar for each assembly, whatever the position of the assembly in the core.* But is it still the case in this reactor, especially for the peripheral blocks which are subjected to a strong variation of the flux and to a very different spectrum compared to the middle of the annular core? The following sections present some results which, along with the results already shown, can be used to

begin answering this question. More studies on realistic cores should be done as well in further studies to provide a more complete answer.

4.7.2 Possible depletion procedures

The interest of doing depletion in the lattice code is that it provides a fine spatial resolution of the flux, and also to perform, for example, a detailed pin power reconstruction once the average power on an assembly has been calculated by the full-core software. However, as was shown in the previous sections (see figures 4.1 and 4.2(a)), the flux can have a very peaked shape in the peripheral assemblies due to the reflector effect. Considering this, the isotopic concentration will not evolve equally everywhere in the core. Therefore, two solutions may be considered:

- the depletion may be partially performed in the core to calculate the new number densities using the flux coming from the full-core calculation and therefore taking all the environment into account. The microscopic cross-section update would still be performed somehow by the lattice code.
- the depletion is performed in the lattice code, which provides macroscopic cross-sections for the full-core calculation. The full-core calculation would only give the burnup of each assembly at each burnup iteration, based on the full-core power map, so that the solver can recover the corresponding cross-sections in the reactor database. This solution, adopted for light water reactors, could work for this reactor only if a way is found to obtain the correct shape of the flux at the lattice level for each type of block in the core. Otherwise, it is likely that the spectral heterogeneity will not be taken into account and thus there would be unacceptable discrepancies after some number of burnup steps.

In the first case, the in-core depletion would provide a very accurate isotopic concentration at each burnup step, especially if the discretization in the peripheral fuel blocks is finer than the typical distance of variation of the flux. This is a very interesting technique regarding the current capabilities employed at the INL because it only requires the implementation of a depletion module in INSTANT and a way to associate its own set of cross-sections for every core region that will be considered as a homogeneous mixture and depleted as well. There would still be the issue of generating cross-sections, because the changes in the concentrations in the core will affect the microscopic cross-sections.

A typical scheme may consist of performing a single block calculation in DRAGON with its own burnup loop. The code would then generate sets of cross-sections tabulated as a function of the burnup of the block (for example) and stored in a library that INSTANT can access. Then in INSTANT, each burnup iteration would be followed by a call to the cross-

section storage so that microscopic cross-sections corresponding to the local burnup of each block are provided in INSTANT. An interpolation module is needed to calculate microscopic cross-sections for burnup step that may not fall on the burnup points of the library. The macroscopic cross-section would then be calculated using the number densities of each region in the core.

This solution may be accurate as long as a sufficient number of groups are used so that it compensates for:

- the fact that the microscopic cross-sections may not have been calculated with the exact composition of the blocks at a given burnup step;
- and the fact that the homogenization is not performed with a representative flux for peripheral blocks, as has been shown in the preceding sections. This difference concerns both flux magnitude and flux spectrum. A trial with supercells may alleviate this concern if the calculation time remains manageable.

Another possible solution is a complete depletion in the lattice code for both densities and microscopic cross-sections. To do this and produce meaningful results would mean recreating the spectral environment of every block. One idea to achieve that is to impose the net currents calculated in the full core calculation at the boundaries in the lattice code. This would then avoid the problem mentioned above; the cross-sections would be calculated both with the right shape of the flux and the right composition. It may lead to a gain in accuracy, but the implementation might be difficult and some theoretical work has to be pursued on the issues it presents, for example abandoning the fundamental mode assumption and the necessity of flux unfolding in angle and energy. Such a task is naturally beyond the scope of this thesis.

The goal here is to assess the difference between performing depletion calculations on a single block and on a supercell in DRAGON. This will indicate how the reflector affects the number densities in the block of interest, and how the neutron spectrum changes during depletion.

4.7.3 Settings for depletion

The depletion was performed with following settings for DRAGON: `SYBILT:` calculation in 295 groups in both cases (single block calculation and supercell calculation) using the same sets of homogenized cross-sections as before for the initial calculation, along with the same settings for the spatial discretization of the geometry.

The following time step scheme was adopted for the depletion calculations. Detailed analysis was not performed to evaluate these, as the aim was to evaluate the differences between domain size only:

- 10 day increment from 0 to 50 days ;

– 50 day increment from 50 to 500 days ;
– 100 day increment from 500 to 1000 days.

The power was constant and fixed at 600 MW.tonne^{-1} of initial heavy metal. This corresponds to the specification of this reactor. The fuel is meant to pass three times inside the reactor, so that the very high burnup of 600 GW day.tonne^{-1} is attained.

4.7.4 Method of investigation

To analyse the results of the two methods of depleting the fuel, number densities along a line across the fuel block of interest were examined. This tells whether the shape of the flux has a significant effect on the isotopic variation. Blocks were also homogenized to get the average number densities for each isotope of interest. Finally, the neutron spectra were examined in different locations in a supercell and its change with depletion.

The isotopes selected are ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{242m}Am, ^{241}Am, and ^{243}Am. These isotopes are present in the initial loading.

An initial investigation was performed to determine the importance of performing self-shielding calculations at each burnup step in the depletion. Ordinarily, the self-shielding calculations would be repeated at each burnup step, but this was not possible in the supercell due to the necessity to use pre-homogenized cross-sections as described in the previous sections.

A way to remain consistent between both paths and to circumvent this problem is to suppress the self-shielding in all but the zero-burnup step in both supercells and single block models. The error may not be so important because luckily, the fuel already contains the most important plutonium isotopes, so that their cross-sections will already be self-shielded. The variation of the number densities might not affect greatly this self-shielding. The importance of this approximation has been studied over one single cell.

4.7.5 Study over one cell: importance of self-shielding during depletion

In order to quantify the error committed by not calculating the self-shielding after each burnup iteration, a simple test was performed over one single hexagonal fuel cell. In the trial, the calculation starts with a regular library of 295 groups containing the usual nuclides of the fuel. One cell was burned with a self-shielding calculation after each burnup step, whereas the other was burnt keeping the original self-shielding. The K_{eff} obtained in DRAGON at each time step with and without self-shielding were compared and the final isotopic concentration of several isotopes was evaluated.

Figure 4.13 shows the differences on the K_{eff} as a function of time. The relative difference

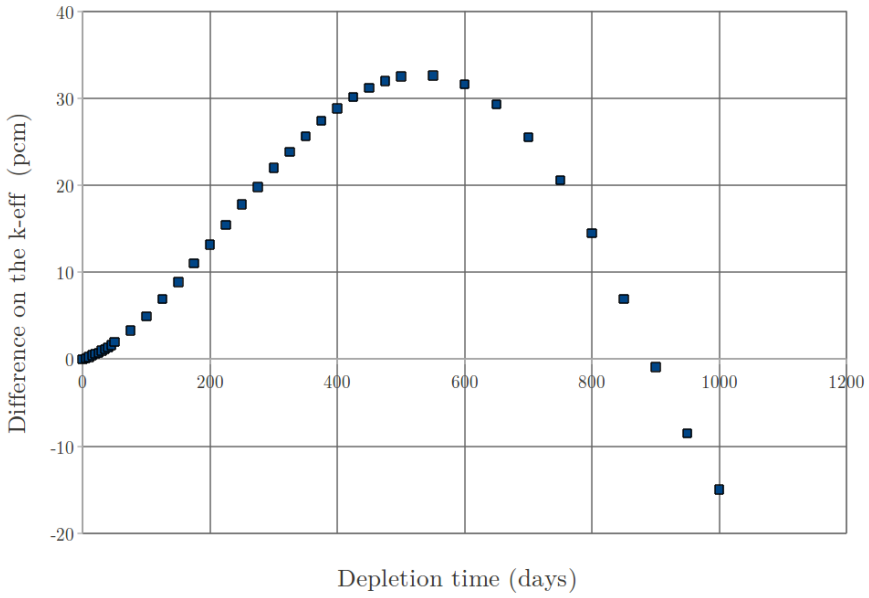


Figure 4.13 – Difference of K_{eff} obtained with and without self-shielding over one fuel cell with transuranic fuel (Plutonium, Americium)

is calculated by the formula:

$$\text{Difference} = \frac{K_{\text{eff without self-shielding}} - K_{\text{eff with self-shielding}}}{K_{\text{eff with self-shielding}}} \times 10^5 \quad \text{(pcm)}. \tag{4.6}$$

This trial shows that the presence or absence of self-shielding during the burnup iteration is responsible for differences of around ± 30 pcm on the K_{eff} in DRAGON and a maximum of $\pm 0,4$ % of differences on the isotopic concentration after 1000 days of depletion. Those relatively good results are due to the fact that the plutonium and americium isotopes were already present at the beginning and thus underwent a self-shielding before entering the burnup loop. The isotopic variation therefore may not affect greatly the self-shielding.

Notice that in the case where some new heavy, transuranic isotopes are created while burning, this assumption may not be valid any more. The calculation was repeated starting only with U^{235} and U^{238} in the fuel instead of the usual Pu-Am fuel, and the results are much less attractive: ± 200 pcm of differences on the K_{eff} (see figure 4.14) are recorded depending whether or not the self-shielding was repeated at each depletion step. The isotopic concentrations differ by ± 20 %. It clearly shows that the newly-created fissile isotopes need to be self-shielded at each time step. After the new nuclides are built up, the self-shielding update with the burnup may become less important.

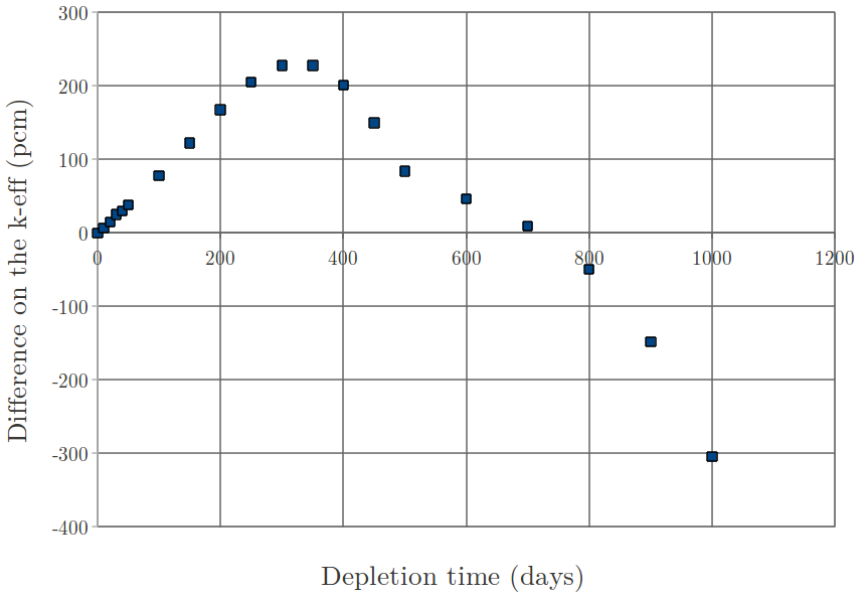


Figure 4.14 – Difference of K_{eff} obtained with and without self-shielding over one fuel cell with a conventional fuel (uranium only)

4.7.6 Isotopic depletion across the block of interest

Single block path

Figure 4.15 shows the evolution of the K_{eff} during the depletion from $t = 0$ days to $t = 1000$ days.

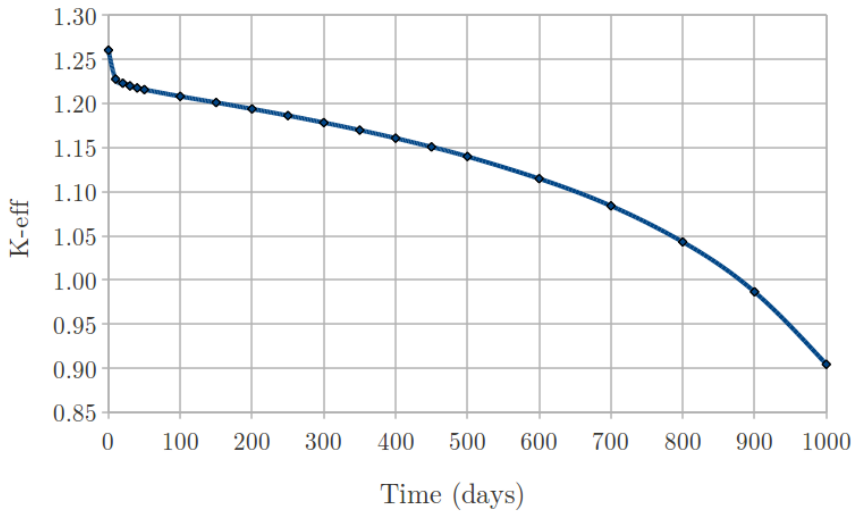


Figure 4.15 – Evolution of the K_{eff} of the single block during depletion

At the beginning, there is the typical steep gradient on the K_{eff}. Then the evolution

is smoother. Table 4.33 gives the correspondence between the time and the burnup for the block. They are related by a linear relation because the specific power is assumed to be constant during all the cycle. As the specific power is the same in the supercells, the correspondence will also be the same.

Table 4.33 – Single block - Time-Burnup equivalence

Time (days)	0	10	20	30	40	50	100
Burnup (GW day.tonne^{-1})	0	6	12	18	24	30	60
Time (days)	150	200	250	300	350	400	450
Burnup (GW day.tonne^{-1})	90	120	150	180	210	240	270
Time (days)	500	600	700	800	900	1000	
Burnup (GW day.tonne^{-1})	300	360	420	480	540	600	

The final burnup (600 GW day.tonne^{-1}) is much higher than what is currently reached in LWRs, but the TRISO particles may allow such a deep burn. It is one of the major interests of this reactor.

The number densities for key heavy isotopes were reported at 0 days, 50 days, 500 days and 1000 days, calculated in various locations. Table 4.34 describes how the fuel is depleting in a fuel cell near the center of the block, next to the central graphite cells.

Table 4.34 – Single block - Variation of the concentration at different time-steps near the center

Isotopes	Relative variation compared to the initial value (%)			
	t = 0 days	t = 50 days	t = 500 days	t = 1000 days
^{237}Np	0,0	-2,9	-28,2	-59,5
^{238}Pu	0,0	4,7	59,6	51,5
^{239}Pu	0,0	-7,9	-65,0	-94,8
^{240}Pu	0,0	-1,3	-25,8	-74,8
^{241}Pu	0,0	10,4	48,7	-37,8
^{242}Pu	0,0	1,9	41,1	116,1
^{241}Am	0,0	-5,8	-45,1	-84,5
^{242m}Am	0,0	70,9	72,0	-66,9
^{243}Am	0,0	4,4	49,3	124,9

It can be observed that all fissile products initially present are disappearing except the ^{242}Pu whose number density is increasing. Of course, minor actinides, like curium, are created as well, some of which may have significant impact on the fuel cycle, but these have not been reported here. The ^{239}Pu is well consumed. Almost 95 % of the initial mass has been fissioned or transmuted after 1000 days. The reactor seems to be quite efficient for burning ^{239}Pu,

which may be of interest for non-proliferation concerns as well as to reduce the amount of waste.

The question is now to determine how homogeneous is the depletion in the single block. For the work on the single block, a fuel cell near the center of the block and a fuel cell in the middle between the center and the edge were considered. This choice is dictated by the fact that the graphite cells located in the center tend to thermalize the neutrons, so that the flux will be different in a fuel cell next to a graphite cell than in a location where the fuel cell is surrounded by fuel cells and coolant cells. This fact will be well shown on figure 4.17 (page 144), which represents the flux condensed to 1 group, plotted across the block from the center to the edge. It will be discussed further in the text in section 4.7.7.

Table 4.35 shows the differences in the number densities at the same time between the two locations of interest inside the single block. Small differences appear between the two locations, which is explained by differences in the flux due to the central graphite cells. The location near the graphite in the center is taken as a reference from which relative differences are calculated.

Table 4.35 – Single block - Variation of the concentration at different time-steps in two locations: near the center (reference) and between the center and the edge

Isotope	Variation of the concentration compared to the center (%)		
	t = 50 days	t = 500 days	t = 1000 days
^{237}Np	0,05	0,49	1,15
^{238}Pu	-0,06	-0,34	0,54
^{239}Pu	0,32	3,49	4,16
^{240}Pu	0,07	2,18	8,77
^{241}Pu	-0,46	-0,99	2,83
^{242}Pu	-0,13	-1,49	-0,91
^{241}Am	0,24	1,89	3,72
^{242m}Am	-0,93	1,80	3,61
Am^{243}	-0,07	-0,94	-1,68

From this, one concludes that even in the single block, the depletion is not totally homogeneous. The maximal difference is of 8,77 % on the ^{240}Pu after 1000 days, which is significant. The density of ^{239}Pu is varying by more than 3,5 % after 500 days and this density is quite important as this is the major fissile isotope in our core, at least during the first part of the irradiation.

Supercell path

Here the same study is performed, but considering a supercell model with 3 reflector blocks and 3 fuel blocks surrounding the block of interest, described previously (see figure

3.2(c) on page 79). Figure 4.16 shows the evolution of the K_{eff} during the depletion from $t = 0$ days to $t = 1000$ days.

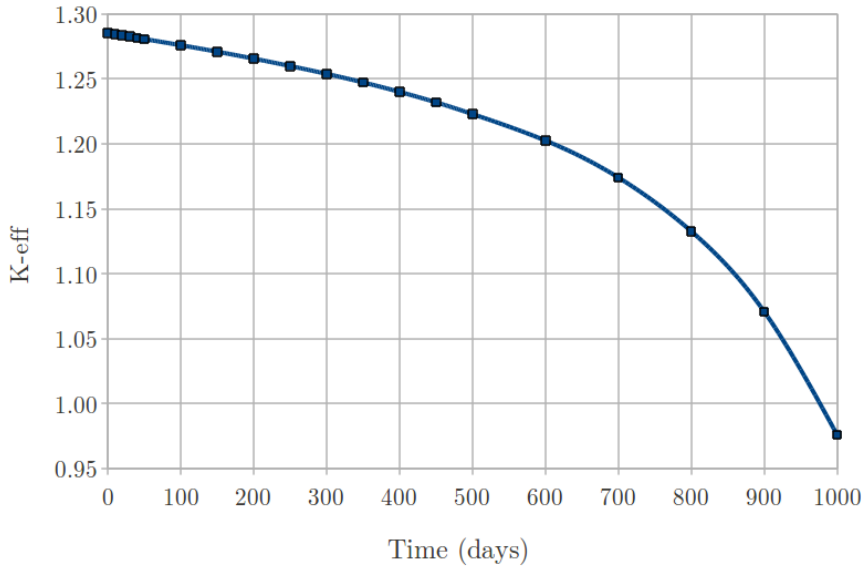


Figure 4.16 – Evolution of the K_{eff} of the supercell model during depletion

The number densities have been recovered at different locations across the block of interest. Values are presented for a fuel cell located at the extreme right, next to the reflector block. As it was shown in [5], the flux spectrum in this region is very different due to the thermalization of the neutrons in the reflector. This is confirmed in section 4.7.7. Differences are therefore expected in the depletion. Table 4.36 gives the variation of the number densities in this right cell compared to the initial values at 0 days, 50 days, 500 days and 1000 days.

Table 4.36 – Supercell - Isotope densities at various times in a fuel cell located next to the reflector

Isotopes	Relative variation compared to the initial value (%)			
	$t = 0$ days	$t = 50$ days	$t = 500$ days	$t = 1000$ days
^{237}Np	0,0	-2,7	-27,2	-58,9
^{238}Pu	0,0	7,4	38,5	-7,3
^{239}Pu	0,0	-11,6	-76,3	-96,5
^{240}Pu	0,0	2,1	-10,9	-60,9
^{241}Pu	0,0	0,1	-11,6	-63,4
^{242}Pu	0,0	6,0	67,9	140,6
^{241}Am	0,0	-6,6	-53,7	-89,6
^{242m}Am	0,0	34,1	-23,5	-84,7
^{243}Am	0,0	2,4	34,1	93,4

Initially, one sees in table 4.36 that the evolution is similar to what was observed with the

single block model. For example, after 1000 days, the ^{239}Pu number density has decreased by 96,5 % in the supercell, and by 94,8 % in the single block. However, differences can already be noticed: for example the ^{238}Pu concentration after 1000 days is equal to -7,3 % after 1000 days in the supercell, while it is equal to +51,5 % of its initial value in the single block.

Table 4.37 provides a comparison of the depletion at time $t = 1000$ days between the cell taken in the single block and the cell taken in the supercell, so that the comparison is more straightforward.

Table 4.37 – Comparison between the fuel cells taken from the single block model and the supercell model after 1000 days irradiation

	Single block (% of the initial density)	Supercell (% of the initial density)	Difference SC-SB (% of the initial density)
^{237}Np	-59,5	-58,9	0,6
^{238}Pu	51,5	-7,3	-58,8
^{239}Pu	-94,8	-96,5	-1,7
^{240}Pu	-74,8	-60,9	13,9
^{241}Pu	-37,8	-63,4	-25,5
^{242}Pu	116,1	140,6	24,6
^{241}Am	-84,5	-89,6	-5,1
^{242m}Am	-66,9	-84,7	-17,8
^{243}Am	124,9	93,4	-31,6

This shows that some of the isotopes, the ^{238}Pu and to a lesser extent ^{241}Pu, do not behave the same way in the two models. In the fuel cell of the single block model, the ^{238}Pu had been only produced during the cycle, while in the supercell, it is produced during the first 500 days but then consumed, ending the irradiation with less than was present initially.

Again, it is of interest to assess whether the depletion is the same between a fuel cell located next to the reflector on the right, and a fuel cell located next to a fuel block (on the left). Therefore, isotopic number densities were reported in these two locations at time $t = 1000$ days. Table 4.38 gives the relative density variations compared to the initial number densities in the two locations of interest.

While the simulations are started with exactly the same concentration in each fuel cell at the beginning, the depletion differs depending on the position inside the block. The effect is amplified compared to the single block model because the amount of graphite seen by the fuel cell on the right is much higher than what is seen on the left. A cell located near a fuel block will not have the same spectrum as a cell located near the reflector, whose presence is drastically affecting the depletion.

On the one hand, the left part seems to behave more or less like what was observed on the single block model. It is not completely equivalent: it might be a bit affected by the

Table 4.38 – Supercell model - Evolution of the number densities in two locations between $t = 0$ days and $t = 1000$ days

Location near a fuel block (left)		Location near a reflector block (right)	
Isotope	Density relative variation (%)	Isotope	Density relative variation (%)
^{237}Np	-57,8	^{237}Np	-58,9
^{238}Pu	42,4	^{238}Pu	-7,3
^{239}Pu	-94,4	^{239}Pu	-96,5
^{240}Pu	-64,0	^{240}Pu	-60,9
^{241}Pu	-46,6	^{241}Pu	-63,4
^{242}Pu	113,8	^{242}Pu	140,6
^{241}Am	-84,6	^{241}Am	-89,6
^{242m}Am	-72,2	^{242m}Am	-84,7
^{243}Am	120,0	^{243}Am	93,4

reflector, but more probably, the different burning state on the right part of the block may lead to different conditions for the left side.

On the other hand, the right side of the block displays very different behaviour than the left. For example, ^{238}Pu is produced on the left, increasing the initial concentration by 42,4 %; but it is consumed on the right, reaching a concentration below the initial one.

Table 4.39 gives the variation of number densities for every isotope at different dates between the left and the right side of the block, the reference being the left side. The discrepancy is becoming larger during the depletion, especially in the middle of the depletion at $t = 500$ days. The ^{239}Pu number density differs by 40 % between the two cells, which is rather important and much higher than what was observed on a single block.

Table 4.39 – Supercell model - Variation of the concentration between locations at different time-steps (%)

Isotope	Concentration's variation between left and right (%)		
	$t = 50$ days	$t = 500$ days	$t = 1000$ days
^{237}Np	-0,4	-3,3	-2,6
^{238}Pu	-0,8	-17,1	-34,9
^{239}Pu	-5,2	-40,3	-38,1
^{240}Pu	3,0	8,8	8,6
^{241}Pu	-7,3	-35,2	-31,4
^{242}Pu	4,3	22,7	12,5
^{241}Am	-2,3	-23,4	-32,8
^{242m}Am	-15,5	-53,1	-45,0
^{243}Am	-1,2	-5,8	-12,1

4.7.7 Spectrum and flux study

Finally, fluxes were recovered in 1 group and the spectra of various locations were compared as described in section 4.5 for the single block and the supercells at $t = 0$ and $t = 1000$ days.

Single block

As would be expected, the flux over the single block is more or less flat, as shown on figure 4.17. This explains why the depletion is quite homogeneous inside the block.

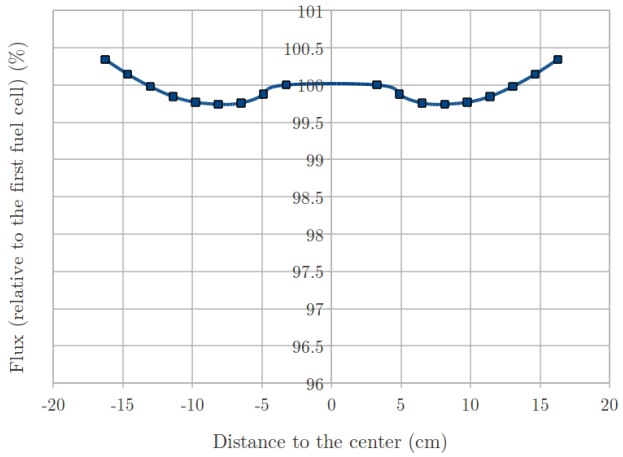


Figure 4.17 – Single block: relative flux in 1 group across the single block, $x = 0$ being the center

Figure 4.18 shows the spectral changes between $t = 0$ days and $t = 1000$ days. Here the spectrum comes from the flux in 295 groups homogenized over the block. There is no significant change in spectrum across the block, explaining why the densities do not differ much between two fuel regions inside the block. While depleting, the spectrum is becoming more thermal, as the thermal cross-sections are reduced and neutrons spend more time thermal before being absorbed.

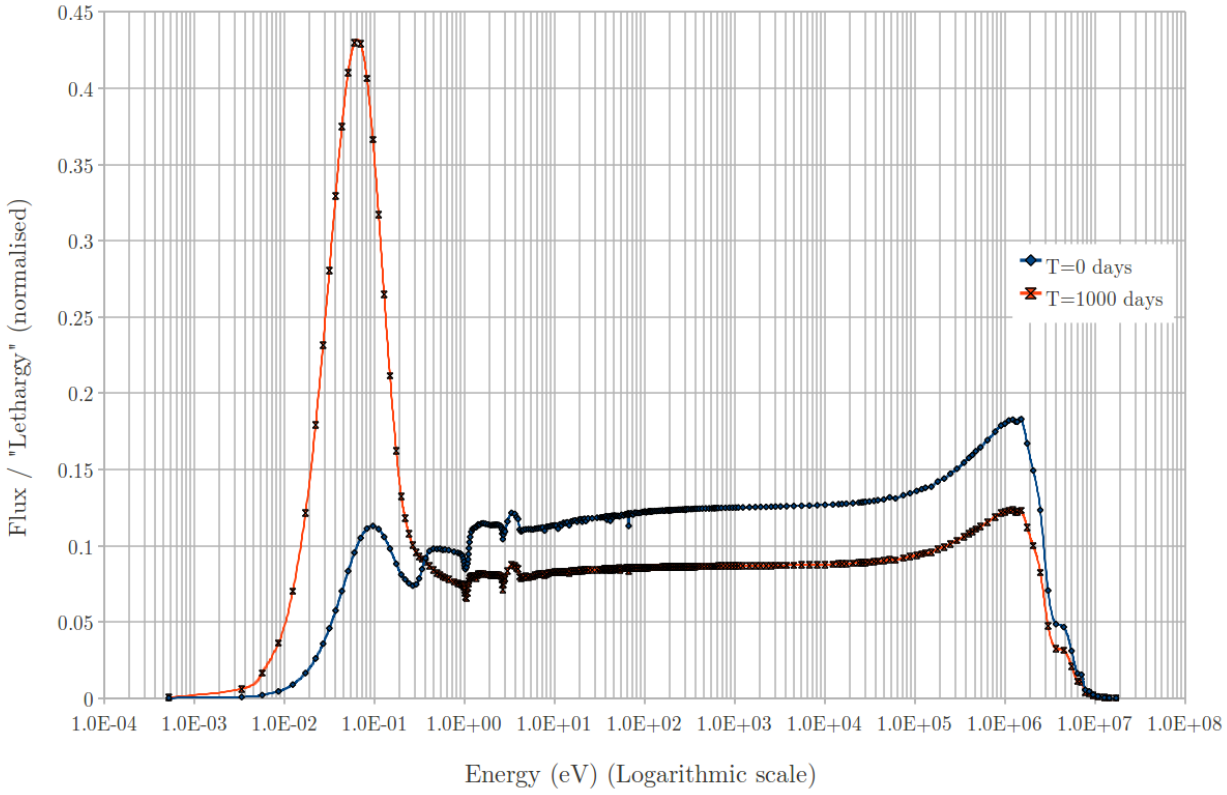


Figure 4.18 – Single block: spectral evolution between $t = 0$ days and $t = 1000$ days

Supercell

For this study, the same supercell with 3 blocks of reflector is considered.

The plot of the flux in 1 group across the block of interest reveals that it cannot be considered as flat (see figure 4.19). With more than 1 group, more variations may be observed.

The magnitude is not the only parameter changing: the spectrum also exhibits strong variations. Figure 4.20 shows the spectra obtained in different fuel cells on a line crossing the center of the block from the left side (next to a fuel block) to the right side (next to a reflector block). The cells were numbered F_1 to F_8, F_1 being the closest cell to the fuel block on the left and F_8 being the closest cell to the reflector block on the right.

The spectra in cells F_1 to F_4, that is to say on the left part of the block of interest, look quite similar. Therefore, they were removed from the plot to clarify the chart. Entering the right part of the block, closer to the reflector, it is observed that the thermal peak's magnitude is increasing. The highest magnitude is reached in fuel cell F_8 next to the reflector. On the contrary, the epithermal flux is higher in the region next to the fuel block because the neutrons coming from the neighbouring fuel blocks are less thermalized. The fast flux is higher in cells located next to the reflector probably because the thermal neutrons coming

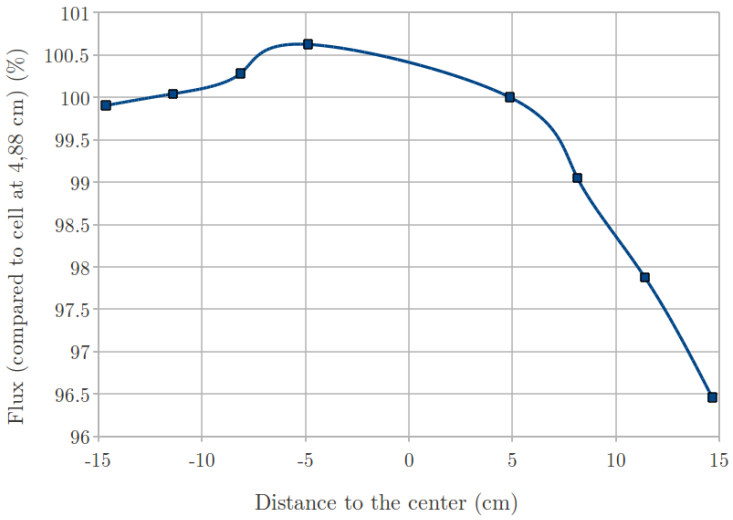


Figure 4.19 – Supercell: flux in 1 group across the block, $x = 0$ being the center

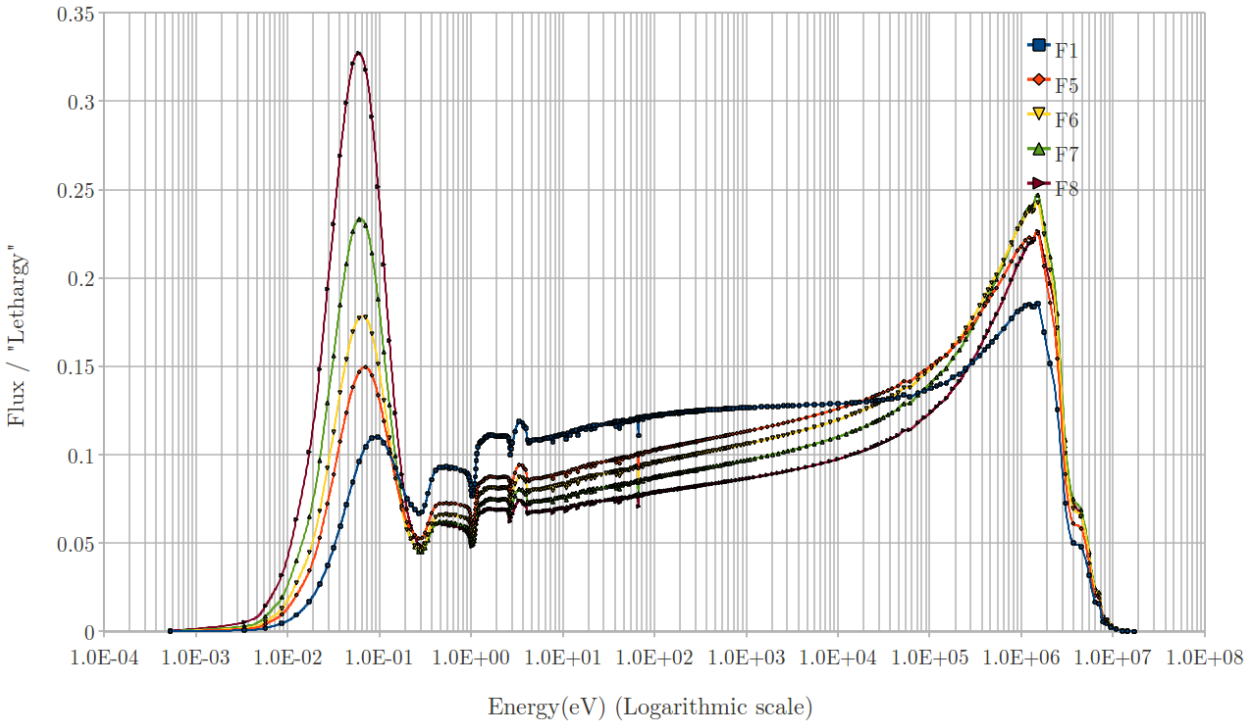


Figure 4.20 – Supercell: neutron energy spectra in different regions of the block of interest, $t = 0$ days

back from the reflector are well thermalized and directly absorbed in the first fuel regions, leading to fissions and thus to the release of fast neutrons. A good part of those neutrons may enter the reflector, be scattered and come back with thermal energies to the fuel block, so that the thermal flux is increased.

This spectral variation will strongly affect the depletion because the thermal neutrons coming from the reflector will have a higher probability of being captured. This explains, for example, why the ^{239}Pu is more burned near the reflector: its absorption cross-section is higher at low energies, and this region will receive more neutrons in this range of energies, so that the absorption reaction rate will be higher.

At the end of the depletion, the same pattern is observed (see figure 4.21). The right half of the block exhibits a higher thermal peak. Globally, this peak is higher at the end of the depletion than at the beginning, which is explained by the fact that there is less fissile isotopes and neutrons spend more time thermal before being absorbed. However, note that the difference is less pronounced: at $t = 0$ days, the thermal peak is more than 3 times higher in F_8 compared to F_1; at $t = 1000$ days, it is just 1,4 times higher.

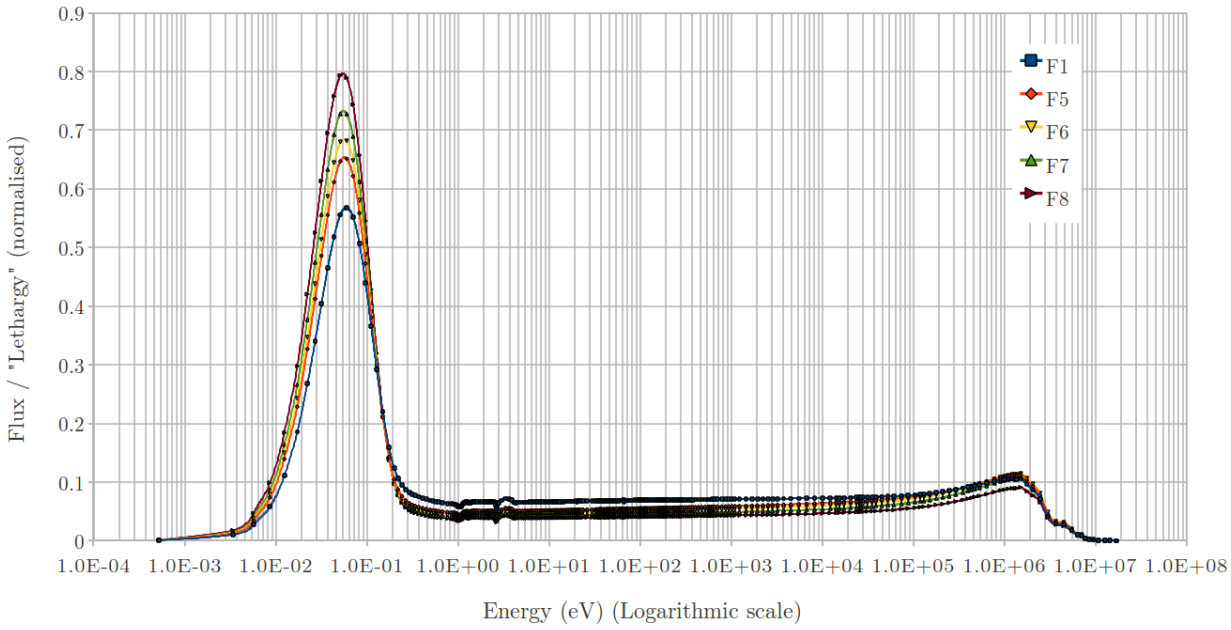


Figure 4.21 – Supercell: neutron energy spectra in different regions of the block of interest, $t = 1000$ days

4.7.8 Conclusion of the depletion study

Several conclusions can be drawn from this study. First, a block surrounded by reflector and fuel blocks exhibits a strong variation in its neutron energy spectrum, with a thermal peak

which is 1,5 to more than 3 times higher on the reflector side than near the neighbouring fuel block. As a consequence, the fuel will not burn in a homogeneous manner. After 1000 days, large discrepancies can be observed regarding the densities between regions next to a fuel block or next to a reflector block. This heterogeneity is much less severe if the depletion is performed over a single block model. Thus, the use of a traditional scheme where the concentrations are calculated in the lattice code seems to be difficult to reconcile with the heterogeneity of the number density evolution.

Therefore, if an in-core depletion model is used, it would be favourable that the block near the reflector is meshed into several small regions were the number densities are calculated using the local flux, to allow calculation of different macroscopic cross-sections for each region. Otherwise, a macroscopic cross-section applied to the entire block and based on an average of the densities may not take into account the high heterogeneity of those blocks. The typical size of an element of the mesh may have to be smaller than the typical distance of variation of the flux to allow one to consider a constant flux over the element while still preserving the overall density changes across a peripheral block.

Besides the heterogeneity of burning inside the core, one should also pay attention to the fact that the burnup of an assembly may not necessarily be a sufficient prediction of nuclides densities or possibly cross-sections. It is very likely that because the peripheral blocks exhibit a different spectrum than the one of the central fuel blocks, there may be differences in the composition at the same burnup. This remark tends to imply the use of supercells to correct the spectrum while the depletion takes place at the lattice level, too.

The choice of not using the supercells may be less attractive once burnable poisons and control rods are added. Moreover, if the blocks are divided into small regions in the core calculation, then it might be interesting to recover microscopic cross-sections corresponding to each small domain using supercells. Such a calculation scheme would certainly require an improvement of the speed of the MOC solver in DRAGON to be considered a viable option.

Given the previous remarks, the solution of coupling the lattice calculation and the core calculation with net currents at the interfaces of each block may greatly improve the fidelity of the simulation and facilitate both depletion analysis and cross-section generation for this reactor by solving directly the two problems. This may be a project worth investigating for future work.

CHAPTER 5

CONCLUSION

5.1 Summary of results

The project of a high temperature reactor pursued by the Idaho National Laboratory requires accurate neutronic simulations of the core behaviour in all situations to meet certification requirements from the safety authorities: with control rods, burnable poisons, depletion of the fuel, and also in accidental conditions where transient analysis must be performed. Within this framework, the influence of the reflector was evaluated in the calculation scheme. The core has been simplified by removing burnable poisons and control rods. Only steady-state calculations were performed and in cold conditions.

Compared to the usual procedures used for light water reactor analysis, two major differences have been identified:

- the annular design of the core leads to a high peaked shape of the flux at both interfaces between fuel and graphite reflector;
- the graphite reflector thermalizes the spectrum, and this spectral variation is noticeable on the first half of the peripheral fuel blocks.

The spectral change as well as the peaked shape of the flux in the half of the fuel adjacent to the reflector raise issues for the lattice physics because a priori, a lattice calculation over a single block may not yield a spectrum representative enough of the real spectrum of the peripheral blocks. To assess that, two paths have been studied to prepare the cross-sections. The first one considered only a single block model at the lattice level, while the second introduced supercells to take into account some of the effects of the reflector. Both calculation paths provide homogenised cross-sections condensed to several group-structures to allow evaluation whether keeping more groups at the core level may compensate for having introduced error in the cross-sections for peripheral blocks at the lattice stage.

The results show that with the method of characteristics in DRAGON, both paths lead to quite good predictions of the Eigenvalue and of the averaged fission rates of each fuel block in the core if enough groups are kept in INSTANT: for example 26 groups. The supercell path produces the best results compared to the equivalent calculations using the single block path. A K_{eff} within ± 20 pcm from MCNP was obtained in both cases. The fission rate errors are within $\pm 1\%$ with the supercells model, and within $[-1\%; +1,88\%]$ with the single block path, compared to the values from MCNP. Calculations with the interface currents method proved

to be less accurate, especially for supercells. But considering the good agreement reached by the single block path with MCNP and considering the significant calculation durations, the use of supercells may not appear as attractive for this simplified core.

Regarding the use of super-homogenisation (SPH) in the supercell calculations, it has been shown that while the Eigenvalues are improved, especially with few groups in the core calculations, it causes to deteriorate the agreement of the fission rate maps with MCNP. This effect has not been fully understood, because SPH ensures that the reaction rates of the heterogeneous lattice calculation are conserved when the same calculation is done with homogeneous cross-sections. It should therefore improve the quality of the cross-sections transferred to the core calculation. The MOC-MOC equivalence may probably not be suitable for a core calculation with INSTANT: the homogeneous solution should be based on the kernel used for the whole core solver.

It is finally noticed that if a more complicated calculation scheme was to be used, for example with one particular supercell calculation for each block, one should consider the development of an integrated environment combining the lattice and core calculations to save time defining geometries and avoiding errors during the transfers of information.

5.2 Limitations of the single block path

Several concerns remain and may play against a calculation using only the single block model.

First, the introduction of burnable poisons and control rods may create geometric difficulties and spectral effects that may not be carried to the core model using a single block model, even with many groups in the core calculation. It is especially important to correctly model the burnable poison effects because their number density variation strongly affects the local flux.

Perhaps still the main challenge remains the depletion calculation in the code. Trials on a supercell have shown that the number densities across a block partially surrounded by reflector vary in significant proportions, up to 45 % between two sides of the block for certain isotopes. In addition, the neutron spectrum is also changing significantly depending on whether the reflector is close or not to the considered fuel cell. Thus, two phenomena are happening in those blocks: a high variation of the fission rates, which tends to create a heterogeneous burning in the block; and a spectral variation which may lead for the same burnup to a different fuel composition from what is achieved in the center or on a single block.

5.3 Perspectives and future research directions

The burnup and composition dependence to the spectrum seems to be the next important feature to be studied to provide scientific basis for deciding what depletion scheme can be used on the VHTR. Adding burnable poisons and control rods to the calculation will allow completion of the study and provide a basis for developing a complete calculation scheme for this reactor. Incidentally, the influence of temperature on cross-sections still has to be captured, even if it is not expected to change the fundamental conclusions of this work.

The path forward toward a complete calculation scheme may require some improvements in DRAGON. The exact modelling of the geometry without the jagged boundary at the periphery of the blocks may be necessary. More generally, the memory issues and long computation times are limitations of DRAGON for use in further studies for the prismatic reactor, especially if supercells prove to be required once the complete calculation scheme is defined. Some improvement of the acceleration techniques in the method of characteristics as well as the parallelization of this method in the code may resolve these issues. Finally, INSTANT is currently missing depletion capabilities and easy transfer of cross-sections with DRAGON. The development of a combined code with shared meshes may be an interesting project as well, especially in the contingency of developing calculation schemes with one dedicated supercell lattice calculation for each block of the core.

REFERENCES

[1] Y. Wang, *INSTANT: User Manual*, Idaho Falls, ID, USA, Aug. 2010.

[2] A. Hébert, "Toward DRAGON Version4," in *Topical Meeting on Advances in Nuclear Analysis and Simulation*, Vancouver, Canada, Sep. 2006.

[3] G. Marleau, A. Hébert, and R. Roy, "A user guide for DRAGON Version4," École Polytechnique de Montréal, Montréal, Canada, Tech. Rep. IGE-294, Oct. 2009.

[4] Wang, Y. and Rabiti, C. and Palmiotti, G., "Krylov Solvers Preconditioned With the Low-Order Red-Black Algorithm for the P_N Hybrid FEM for the INSTANT Code," in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, May 2011, (accepted).

[5] M. A. Pope, J. Ortensi, and A. M. Ougouag, "Investigation of Supercells for Preparation of Homogenized Cross-Sections for Prismatic Deep Burn VHTR Calculations," in *Proceedings of HTR 2010*, Prague, Czech Republic, Oct. 2010.

[6] A. Hébert, *Applied Reactor Physics*. Montréal, Canada: Presses internationales Polytechnique, 2009.

[7] P. Reuss, *Précis de neutronique*. Les Ulis, France: EDP Sciences, 2003.

[8] L. Massimo, *Physics of high-temperature reactors*. Pergamon Press, Jan. 1976.

[9] J. Basdevant and J. Dalibard, *Mécanique quantique*. Palaiseau, France: Éditions de l'École Polytechnique, 2002.

[10] M.-N. Sanz and B. Salamito, *Physique tout-en-un PC-PC**. Paris, France: Dunod, 2009.

[11] R. Le Tellier, "Développement de la méthode des caractéristiques pour le calcul de réseau," Ph.D. dissertation, École Polytechnique de Montréal, Montréal, Canada, Dec. 2006.

[12] D. Rozon, *Introduction à la cinétique des réacteurs nucléaires*. Montréal, Canada: Presses internationales Polytechnique, 1992.

[13] A. Hébert, "Development of the Subgroup Projection Method for Resonance Self-Shielding Calculations," *Nuclear Science and Engineering*, vol. 162, no. 1, pp. 56–75, 2009.

[14] M. Livolant and F. Jeanpierre, "Auto-protection des résonnances dans les réacteurs nucléaires. Application aux isotopes lourds." Commissariat à l'Énergie Atomique, France, Saclay, France, Tech. Rep. CEA-R-4533, 1974.

[15] D. E. Cullen, "Application of the probability table method to multigroup calculations of neutron transport," *Nuclear Science and Engineering*, vol. 55, no. 4, pp. 387–400, 1974.

[16] P. Ribon and J.-M. Maillard, "Les tables de probabilité. Application au traitement des sections efficaces pour la neutronique." France, 1986, note CEA-N-2485.

[17] K. S. Smith, "Spatial homogenization methods for light water reactor analysis," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, Jun. 1980.

[18] G. I. Bell and S. Glasstone, *Nuclear reactor theory.* Van Nostrand Reinhold Co., 1970.

[19] L. J. Cox et al., "MCNP Version 5," in *Proc. 12th Biennial Topl. Mtg.*, Santa Fe, NM, USA, Apr. 2002.

[20] J. Leppännen, "Development of a New Monte Carlo Reactor Physics Code," Ph.D. dissertation, Helsinki University of Technology, Helsinki, Finland, 2007.

[21] N. Hfaiedh and A. Santamarina, "Determination of the Optimized SHEM Mesh for Neutron Transport Calculations," in *Proc. Topl. Mtg. Mathematics and Computations, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, Avignon, France, Sep. 2005.

[22] N. Hfaiedh, "Nouvelle Méthodologie de Calcul de l'Absorption Résonnante," Ph.D. dissertation, Université Louis Pasteur, Strasbourg, France, Sep. 2006.

[23] N. Hfaiedh and A. Santamarina, "Refinement of the Santamarina-Hfaiedh Energy Mesh between 22.5 eV and 11.4 keV," in *Proc. Int. Conf. Physics of Reactors*, Interlaken, Switzerland, Sep. 2008.

[24] Y. Wang, *Equations of P_N-hybrid-FEM for the multigroup transport equation*, Idaho Falls, ID, USA, Jul. 2010.

[25] C. H. Lee, Z. Zhong, T. A. Taiwo, W. S. Yang, M. A. Smith, and G. Palmiotti, "Status of Reactor Physics Activities on Cross Section Generation and Functionalization for the Prismatic Very High Temperature Reactor, and Developpement of Spatially-Heterogeneous Codes," Argonne National Laboratory, Argonne, IL, USA, Tech. Rep. ANL-GenIV-075, Aug. 2006.

[26] C. H. Lee, Z. Zhong, T. A. Taiwo, W. S. Yang, H. S. Khalil, and M. A. Smith, "Enhancement of REBUS-3/DIF3D for Whole-Core Neutronics Analysis of Prismatic Very High Temperature Reactor (VHTR)," Argonne National Laboratory, Argonne, IL, USA, Tech. Rep. ANL-GenIV-076, Sep. 2006.

[27] T. A. Taiwo and T. K. Kim, "Evaluation of the DRAGON Code for VHTR Design Analysis," Argonne National Laboratory, Argonne, IL, USA, Tech. Rep. ANL-GenIV-060, Sep. 2005.

[28] J. Ortensi, J. Cogliati, M. Pope, J. Bess, R. Ferrer, A. Bingham, and A. Ougouag, "Deterministic Modeling of the High Temperature Test Reactor," Idaho National Laboratory (INL), Idaho Falls, ID, USA, Tech. Rep., 2010.

[29] A. Hébert, "A Collision Probability Analysis of the Double-Heterogeneity Problem," *Nuclear Science and Engineering*, vol. 115, no. 2, p. 177, Oct. 1993.

[30] A. Hébert, "Scattering Reduction of the Double-Heterogeneity Treatment in Dragon," *Nuclear Science and Engineering*, vol. 160, no. 2, pp. 261–266, Oct. 2008.

[31] M. Ouisloumen, "Résolution par la méthode des probabilités de collision de l'équation intégrale du transport à deux et trois dimensions en géométrie hexagonale," Ph.D. dissertation, École Polytechnique de Montréal, Montréal, Canada, Oct. 1993.

[32] A. Hébert, "Mixed-dual implementations of the simplified P_N method," *Annals of Nuclear Energy*, vol. 37, no. 4, pp. 498–511, Apr. 2010.

[33] A. Henry, *Nuclear-reactor analysis*. Cambridge, MA, USA: MIT press, 1975.

[34] IAEA, "Evaluation of high temperature gas cooled reactor performance: Benchmark analysis related to initial testing of the HTTR and HTR-10," IAEA, Vienna, Austria, Tech. Rep. IAEA-TECDOC-1382, Nov. 2003.

[35] M. Schmidt-Hoenow and R. Rucker, "Analysis of VHTRC Physics Benchmarks," Argonne National Laboratory, Argonne, IL, USA, Tech. Rep. DOE-HTGR-90363, Rev. 0, May 1993.

APPENDIX A

Geometric relations in a hexagon

Useful geometric relations for hexagons follow [1]:

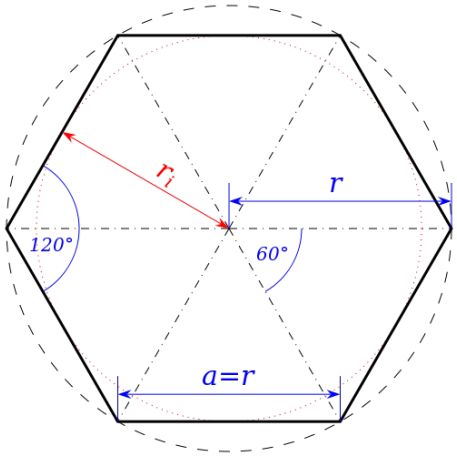


Figure A.1 – Geometric relations in a hexagon

$$a = r \tag{A.1}$$

$$r_i = a\frac{\sqrt{3}}{2} \tag{A.2}$$

$$\text{Pitch: } p = 2\,r_i \tag{A.3}$$

$$\text{Area: } \mathscr{A} = 3\frac{\sqrt{3}}{2}a^2 \tag{A.4}$$

1. The figure was taken from Wikipedia : `http://fr.wikipedia.org/wiki/Hexagone`

APPENDIX B

Formula to calculate the number of hexagons in a $1/12^{\text{th}}$ core

Say that N is the number of hexagons on a row from the center to the periphery of a hexagonal array of hexagons and T is the total number of hexagons on the $1/12^{\text{th}}$ geometry.

First case: suppose N odd:

$$\exists p \in \mathbb{N} \ / \ N = 2p + 1$$

$$\Rightarrow \quad T = \sum_{i=0}^{p}(2i+1) = \sum_{i=0}^{p} 1 + 2\sum_{i=0}^{p} i = p + 1 + 2\frac{p(p+1)}{2} = p^2 + 2p + 1 = (p+1)^2 \quad \text{(B.1)}$$

Then replacing p in the previous formula using $p = \frac{N-1}{2}$, we find:

$$T = \left(\frac{N-1}{2} + \frac{2}{2}\right)$$

$$\Leftrightarrow \quad T = \frac{(N+1)^2}{4} \quad \text{with N odd.} \quad \text{(B.2)}$$

Second case: suppose N even:

$$\exists p \in \mathbb{N} \ / \ N = 2p$$

$$\Rightarrow \quad T = \sum_{i=0}^{p}(2i) = p(p+1) \quad \text{(B.3)}$$

Then replacing p in the previous formula using $p = \frac{N}{2}$, we find:

$$T = \frac{N}{2}\left(\frac{N}{2} + 1\right) = \frac{N^2}{4} + \frac{N}{2} = \frac{N}{4}(N+2)$$

$$\Leftrightarrow \quad T = \frac{N(N+2)}{4} \quad \text{with N even.} \quad \text{(B.4)}$$

APPENDIX C

Homogenized graphite density calculation over a single block

This appendix explains the calculation of the graphite average densities over the real block and over the DRAGON single block model. This is necessary to determine the density of the graphite in the J cells which alleviates the lack of a proper geometry in DRAGON, but also to build correct macroscopic cross-sections for the graphite in INSTANT.

Thanks to the symmetry, the calculation is performed for a $1/12^{\text{th}}$ block. This allows comparison of some parts of the calculation with what DRAGON is calculating by itself. Thus, the geometry contains:

- 17,5 F cells
- 8,5 C cells
- $\frac{1}{2}$ K cell
- $1+\frac{1}{12}$ G cell
- the boundary area S_b or 5,5 J cells, depending of the model.

The normal graphite density d_{GRA} is set to $d_{GRA} = 8.774.10^{-2}.10^{24}$ atoms.cm^{-3}. Basically, the number of atoms and the global area for each geometry are computed, along with the homogenized surface densities in both geometries. The density in the J cells is then adjusted depending on what need to be conserved: the number of atoms, or the global average graphite density.

C.1 Internal domain: 10 rings

This domain is the same for the two geometries. For a better understanding, it is first supposed that the cells are prismatic (3D) and not hexagonal (2D). We define "h", the height of a hexagonal prism.

C.1.1 F cells

The F cells are composed of an internal tube pellet with a matrix filled with TRISO particles and graphite, an annular region filled with helium and a surrounding region of graphite. The following notations are used:

- R_1 and R_2: the radii of the embedded tubes.
- $\mathcal{A}$: the area of the hexagonal basis
- a: the side of the hexagonal basis

– η: the packing fraction
– V_T: the volume occupied by all TRISO particles
– V_R: the volume of the matrix.

The packing fraction corresponds to the volume fraction of TRISO particles present in the matrix volume. We have: $\eta = V_T/V_R$.

A TRISO particle is made up of a sphere with 5 layers having the following radii and with following graphite densities:

Radius (cm)	Graphite volume density ($.10^{24}$ atoms.cm^{-3})
$r_1 = 0,01$ cm	$d_{GRA1} = 0,0$
$r_2 = 0,022$ cm	$d_{GRA2} = 5,265.10^{-2}$
$r_3 = 0,026$ cm	$d_{GRA3} = 9,526.10^{-2}$
$r_4 = 0,0295$ cm	$d_{GRA4} = 0,0$
$r_5 = 0,0335$ cm	$d_{GRA5} = 9,526.10^{-2}$

The surrounding matrix has a density of $d_{GRA-F} = 8,524.10^{-2}.10^{24}$ atoms.cm^{-3}.

Let N be the number of TRISO particles contained in the matrix. The packing fraction is defined by:

$$\eta = \frac{\frac{4}{3}\pi r_5^3 \cdot N}{\pi R_1^2\ h}$$

Thus we have:

$$N = \eta \cdot \frac{\pi R_1^2\ h}{\frac{4}{3}\pi r_5^3} \text{: number of TRISO particles in the matrix}$$

Let N_T be the total number of graphite atoms in a single TRISO particle.

$$N_T = \frac{4}{3}\pi \left(r_2^3 - r_1^3\right) d_{GRA2} + \frac{4}{3}\pi \left(r_4^3 - r_3^3\right) d_{GRA3} + \frac{4}{3}\pi \left(r_5^3 - r_4^3\right) d_{GRA5}$$

The total number of graphite atoms N_F contained in an F cell is composed of three terms: the graphite atoms contained in all the TRISO particles, those contained in the matrix, and those located in the external surrounding volume.

$$N_F = \eta \cdot \frac{\pi R_1^2\ h}{\frac{4}{3}\pi r_5^3} \cdot N_T + (1-\eta) \cdot \pi R_1^2\ h \cdot d_{GRA-F} + \left(\mathcal{A} - \pi R_2^2\right)\ h \cdot d_{GRA}$$

The height h can be factored out:

$$N_F = h \cdot \left(\eta \cdot \frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3} \cdot N_T + (1-\eta)\,\pi R_1^2 \cdot d_{GRA-F} + (\mathcal{A} - \pi R_2^2) \cdot d_{GRA}\right)$$

Finally N_{totF} is the total number of atoms contained in all F cells of the geometry:

$$N_{totF} = 17,5\ N_F$$

C.1.2 C cells

These cells contain a tube filled with helium surrounded by graphite with the normal density d_{GRA}. Calling R the radius of the embedded tube, and N_C the number of graphite atoms in one cell of this type. $R = 0,794$ cm.

$$N_C = h \cdot \left(\mathcal{A} - \pi R^2\right) \cdot d_{GRA}$$

Calling N_{totC} the total number of atoms contained in all C cells of our geometry.

$$N_{totC} = 8,5\ N_C$$

C.1.3 K cells

These cells contain a tube filled with helium surrounded by graphite with the normal density d_{GRA}. Calling r the radius of the embedded tube, and N_K the number of graphite atoms in one cell of this type. $r = 0,635$ cm.

$$N_K = h \cdot \left(\mathcal{A} - \pi r^2\right) \cdot d_{GRA}$$

Calling N_{totK} the total number of atoms contained in all K cells of the geometry:

$$N_{totK} = 8,5\ N_K$$

C.1.4 G cells

These cells are simple hexagons with normal graphite density. Calling N_G and N_{totG} the number of graphite atoms in one G cell and in all G cells of our geometry:

$$N_G = h \cdot \mathcal{A} \cdot d_{GRA}$$
$$N_{totG} = (1 + \frac{1}{12})\ N_G$$

It is then possible to compute the total number of graphite atoms located in the internal prism N_{int}:

$$N_{int} = N_{totF} + N_{totC} + N_{totK} + N_{totG}$$

Note that "h" is in factor in front of every term.

C.2 Boundary region and global volume

C.2.1 DRAGON model

The model use 5,5 J cells with adjusted graphite density d_J that form the boundary region. The total number of atoms contained in the J cells is calculated as follows:

$$N_{totJ} = 5,5\ h \cdot \mathcal{A} \cdot d_J$$

Thus the total number of atoms in the DRAGON model is equal to:

$$N_{DRAGON} = N_{int} + 5,5\ h \cdot \mathcal{A} \cdot d_J$$

To calculate the global volume in the DRAGON model, the number of cells are summed, and this is multiplied by the area of a hexagon $\mathcal{A}$. Calling S_{DRAG} the area of the base of the prismatic block and V_{DRAG} the volume of the prismatic block, we have:

$$S_{DRAG} = (17,5 + 8,5 + 0,5 + 1 + \frac{1}{12} + 5,5) * \mathcal{A}$$
$$V_{DRAG} = h \cdot S_{DRAG}$$

Again, "h" is in factor in V_{DRAG}.

It is now possible to calculate the density for the DRAGON model:

$$d_h = \frac{N_{DRAG}}{V_{DRAG}}$$

As shown previously, h is in factor in N_{DRAG} and V_{DRAG}. It will be simplified by the division, which leads to this quite important conclusion: in the case of a tube pellet with spherical particles, the global homogenized density is independent from the height h of the tube.

This is why it is possible to redefine everything in terms of numbers of atoms over an hexagon: one can divide the previous quantities of atoms by h.

$$N_T = \frac{4}{3}\pi\left(r_2^3 - r_1^3\right)\cdot d_{GRA2} + \frac{4}{3}\pi\left(r_4^3 - r_3^3\right)\cdot d_{GRA3} + \frac{4}{3}\pi\left(r_5^3 - r_4^3\right)\cdot d_{GRA5}$$

$$N_{totF} = 17,5\ \left(\eta\cdot\frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3}\cdot N_T + (1-\eta)\,\pi R_1^2\cdot d_{GRA-F} + \left(\mathcal{A} - \pi R_2^2\right)\cdot d_{GRA}\right)$$

$$N_{totC} = 8,5\ \left(\mathcal{A} - \pi R^2\right)\cdot d_{GRA}$$

$$N_{totK} = \frac{1}{2}\ \left(\mathcal{A} - \pi r^2\right)\cdot d_{GRA}$$

$$N_{totG} = \left(1 + \frac{1}{12}\right)\mathcal{A}\cdot d_{GRA}$$

$$N_{int} = N_{totF} + N_{totC} + N_{totK} + N_{totG}$$

$$N_{totJ} = 5,5\ \mathcal{A}\cdot d_J$$

$$N_{DRAG} = N_{int} + N_{totJ}$$

This amounts to say that all densities are meant as surface densities instead of volume densities. Thus the above formulae are homogeneous. N_T becomes the number of atoms of graphite per height in one single TRISO particle because the d_{GRA2}, d_{GRA3} and d_{GRA5} become surface densities. η remains the volumetric packing fraction, without dimension. Its definition is no more straightforward because the volumes of the pellet and of the TRISO spheres are no more defined over a plane surface. In fact, the volume of the TRISO is projected on the plane surface.

The term $\eta\cdot\frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3}\cdot N_T$ has no dimension and can be interpreted as the average number of graphite atoms contained in the TRISO particles and located on a right plane section of the tube. It becomes clearer when it is completely written:

$$\eta\cdot\frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3}\cdot N_T = \eta\cdot\pi R_1^2\ \sum_{i=1}^{5}\left(\underbrace{\frac{\frac{4}{3}\pi\left(r_i^3 - r_{i-1}^3\right)}{\frac{4}{3}\pi r_5^3}}_{\phi_i}\cdot d_{GRAi}\right)$$

where ϕ_i is the volumetric fraction of the layer i in the TRISO particle.

The term in the sum corresponds to the average number of atoms of graphite of a TRISO particle per square centimetre. It can be seen as the average number of atoms counted on any section of a sphere. Thus, multiplied by πR_1^2, it becomes the maximum number of atoms that could be counted on average on a right plane section of the pellet if the pellet were totally filled with TRISO particles. This enables one to give a sort of definition of η in two dimensions: η would be the ratio between the number of graphite atoms belonging to TRISO

particles located on a section, divided by the maximum number of atoms that could be placed on the same section.

$$\eta = \frac{\text{True number of graphite atoms belonging to particles over a section}}{\underbrace{\pi R_1^2}_{\text{Area of the plane right section}} \cdot \underbrace{\sum_{i=1}^{5} \left(\frac{\frac{4}{3}\pi \left(r_i^3 - r_{i-1}^3 \right)}{\frac{4}{3}\pi r_5^3} \cdot d_{GRAi} \right)}_{\text{Average number of graphite atoms.cm}^{-2} \text{ in 1 particle}}}$$

The numerical calculation gives:

$$\begin{aligned}
N_T &= 9,650.10^{-6}.10^{24} \text{ graphite atoms in 1 TRISO particle.} \\
N_{totF} &= 4,481.10^{24} \text{ graphite atoms in all F cells.} \\
N_{totC} &= 0,806.10^{24} \text{ graphite atoms in all C cells.} \\
N_{totK} &= 0,0787.10^{24} \text{ graphite atoms in all K cells.} \\
N_{totG} &= 0,290.10^{24} \text{ graphite atoms in all G cells.} \\
N_{int} &= 5,657.10^{24} \text{ graphite atoms in the internal area.} \\
S_{DRAG} &= 101,264 \text{ cm}^2.
\end{aligned}$$

C.2.2 MCNP model

In the MCNP model, it is easier to compute the boundary area between the internal cells and the frontier by subtracting the internal area from the total area.

Calling S the total area of the large hexagon, $p = 36,0$ the pitch, and s the side of the large hexagon:

$$s = \frac{p}{\sqrt{3}} = 20,784 \text{ cm}^2$$

$$S = \frac{3\sqrt{3}}{2} s^2 = 1122,37 \text{ cm}^2$$

There are 331 internal hexagons, which occupies an area equal to:

$$\mathcal{A}_h = 331 \cdot \frac{\sqrt{3}}{2} (1,88)^2 = 1013,15 \text{ cm}^2$$

Thus, if we call S_b the boundary area for $1/12^{\text{th}}$ of the block:

$$12\ S_b = \mathcal{S} - \mathcal{A}_h = 109,22 \text{ cm}^2$$

Thus the area of the boundary region and the number of atoms in it is found to be:

$$S_b = 9,101 \text{ cm}^2$$

$$N_b = S_b \cdot d_{GRA} = 0,799 \ .10^{24} \text{ graphite atoms.}$$

Finally, the total number of graphite atoms in the MCNP $1/12^{\text{th}}$ block model and the corresponding area are equal to:

$$N_{MCNP} = N_{int} + N_b = 6,455 \ .10^{24} \text{ graphite atoms.}$$
$$S_{MCNP} = \frac{1}{12} \ S = 93,53 \text{ cm}^2.$$

C.3 Conservation of the global homogenized density

In this first example, we are searching d_J so that we keep the same global homogenized surface density d_h in both cases:

$$\frac{N_{MCNP}}{S_{MCNP}} = \frac{N_{DRAG}}{S_{DRAG}} \iff d_J = \frac{1}{5,5 \ \mathcal{A}} \left(\frac{S_{DRAG}}{S_{MCNP}} N_{MCNP} - Nint \right)$$

The numerical application gives:

$$d_J = 7,914.10^{-2}.10^{24} \text{ graphite atoms.cm}^{-2} \Leftrightarrow d_J = 0,902 \cdots d_{GRA}$$
$$d_h = \frac{N_{DRAG}}{S_{DRAG}} = 6,902.10^{-2}.10^{24} \text{ graphite atoms.cm}^{-2}$$

The difference between the global homogenized densities of the MCNP model and the DRAGON model is equal to $1,6.10^{-15}$, which is beyond the precision of DRAGON.

The calculation in DRAGON is consistent with the above calculation: we obtain $d_h = 6,9017820.10^{-2} \ .10^{24}$ graphite atoms.cm^{-2} in DRAGON instead of $d_h = 6,9017826.10^{-2} \ .10^{24}$ graphite atoms.cm^{-2} in the density calculation.

This J-cell density was useful for building directly the right macroscopic cross-sections in DRAGON for INSTANT. The hope was that the different boundaries would not appreciably change the neutron energy spectrum compared to what would give a calculation over the real block. But in fact, in order to provide directly macroscopic cross-sections for INSTANT with the correct densities, all isotope's densities should be adjusted in every region by a factor of about 8,2 %, corresponding to the ratio of $S_{\text{DRAGON}}/S_{\text{MCNP}}$, including the densities of the fissile isotopes. This would significantly change the composition of the fuel and would likely lead to quite wrong cross-sections for INSTANT.

Once the capability was developed to change the density during preparation of the macroscopic cross-sections for INSTANT, this capability was no more of interest. Indeed, the most logical approximation to mimic the real block calculation is to conserve the number of atoms for each isotope between the two block models, to recover the microscopic cross-sections and to establish the new macroscopic cross-sections for INSTANT taking the true densities into account.

The resulting J-cell density has been determined hereafter. Other isotope's densities over a real block have also been determined and are given in next appendix.

C.4 Conservation of the number of graphite atoms in the boundary region

It was preferred to conserve the number of atoms in the boundary regions between the MCNP model and the DRAGON model. This is equivalent to conserving the total number of graphite atoms, because the internal number of atoms is the same in both cases. This choice can be justified by the fact that such a relation would produce a spectrum for the neutrons coming in the boundary fuel rings that should better reflect the spectrum generated in MCNP. This appears to be the most logical approximation to deal with the jagged boundary limitation in DRAGON.

The conservation relation is written as:

$$\begin{aligned} N_{MCNP} = N_{DRAG} &\iff N_{int} + N_b = N_{int} + N_J \\ &\iff N_b = d_J \cdot V_J \\ &\iff d_J = \frac{N_b}{V_J} \end{aligned}$$

We find:

$$d_J = 4,743.10^{-2}.10^{24} \text{ graphite atoms.cm}^{-2}, \text{ or otherwise:}$$
$$d_J = 0,540617 \cdot d_{GRA}$$

This leads to a global homogenized density in the DRAGON model of:

$$d_{h-DRAG} = 6.375.10^{-2} \ .10^{24} \text{ graphite atoms.cm}^{-2}$$

The global homogenized density in MCNP model remains the same:

$$d_{h-MCNP} = 6,902.10^{-2}.10^{24} \text{ graphite atoms.cm}^{-2}$$

So we have:

$$d_{h-DRAG} = 0,9236 \cdot d_{h-MCNP}$$

Thus, the macroscopic cross-sections passed to INSTANT with this method without any other treatment will have lower density than the expected one. Therefore, it necessary to recover the microscopic cross-section from such a calculation, and to calculate the macroscopic cross-section for INSTANT with the right density d_{h-MCNP}.

In the supercell models in DRAGON, each J cell is shared between two blocks, so that the core geometry looks approximately like the real one. Therefore, the J-cell density has to be multiplied by 2 to reflect the right boundary volume. This remark is valid only when trying to conserve the total number of atoms over the whole lattice comparing to MCNP. Therefore, in this case, the following is used:

$d_J = 9,487.10^{-2}.10^{24}$ graphite atoms.cm^{-2}, or otherwise:
$d_J = 1,08123 \cdot d_{GRA}$

Naturally, this leads to a different homogenized density comparing to the MCNP one, but this is not important as only the microscopic cross-sections are recovered from those calculations:

$d_{h-DRAG} = 7,163.10^{-2}.10^{24}$ graphite atoms.cm^{-2}, that is to say:
$d_{h-DRAG} = 1,038 \cdot d_{h-MCNP}$

APPENDIX D

Density calculation for all isotopes in INSTANT

As seen before, the areas of the blocks in INSTANT and DRAGON are different, which leads to different densities and prevents the direct passage of homogenized macroscopic cross-sections computed in DRAGON.

It is necessary to conserve the number of atoms between the two calculations. Only graphite is located in the J-cells. Therefore, for all the other isotopes, the correct way to pass cross-sections to INSTANT is to keep the same number of atoms as in DRAGON, because the inside part of the blocks does not change between the two models.

As the J-cells introduce an extra area at the periphery, the densities calculated by DRAGON will be lower than the correct ones. Thus, they have to be corrected before calculating the macroscopic cross-sections for INSTANT.

The following can be done:

$$d^{isotope}_{INSTANT} = d^{isotope}_{DRAGON} \cdot \frac{S_{DRAGON}}{S_{INSTANT}}$$

For explanation purposes, the full calculation were repeated as for the graphite above, in order to verify the densities calculated in DRAGON. The calculation was done over a $1/12^{\text{th}}$ block and the work done for the graphite was reused. This leads naturally to the same results.

The surface in INSTANT is equal to:

$$S_{INSTANT} = 93,53074 \text{ cm}^2.$$

The model contains 4 kind of isotopes:

– the isotopes located in the kernel of the TRISO particles;
– the isotopes located in the peripheral shells of the TRISO particles;
– the helium located in the annular tube surrounding the fuel pellet and in the coolant channels;
– the graphite located in the TRISO particles, in the matrix of the fuel cells, in the graphite cells and in all areas surrounding coolant channels.

For the isotopes located in the TRISO's kernel, we compute N_T, the number of atoms inside one TRISO particle, then the total number of atoms inside one fuel cell N_{totF}, and finally the homogenized density over the block (see table D.1).

$$\begin{aligned} N_T &= \frac{4}{3}\pi r_1^3 \\ N_{totF} &= 17,5 \times \left(\eta \cdot \frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3} \cdot N_T \right) \\ d_h^{isotope} &= \frac{N_{totF}}{S_{INSTANT}} \end{aligned} \tag{D.1}$$

Table D.1 – Average densities of the fuel isotopes in INSTANT

Isotope	Density $(.10^{24} atoms.cm^{-3})$
O16	$4,703781.10^{-5}$
NP27	$1,613876.10^{-6}$
PU38	$6,854202.10^{-7}$
PU39	$1,165342.10^{-5}$
PU40	$5,390898.10^{-6}$
PU41	$2,053928.10^{-6}$
PU42	$1,138833.10^{-6}$
AM41	$6,535032.10^{-7}$
AM42	$4,649703.10^{-9}$
AM43	$3,241537.10^{-7}$

For isotopes located in the shells of the TRISO particles (see table D.2), the calculation is similar.

$$\begin{aligned} N_T &= \frac{4}{3}\pi\left(r_i^3 - r_{i-1}^3\right) \quad \text{with } i = 4 \text{ for layer 4 and } i = 5 \text{ for layer 5} \\ N_{totF} &= 17,5 \times \left(\eta \cdot \frac{\pi R_1^2}{\frac{4}{3}\pi r_5^3} \cdot N_T\right) \\ d_h^{isotope} &= \frac{N_{totF}}{S_{INSTANT}} \end{aligned} \tag{D.2}$$

Table D.2 – Average densities of the TRISO's shells isotopes in INSTANT

Isotope	Density $(.10^{24} atoms.cm^{-3})$
SI28	$3,779168.10^{-4}$
SI29	$1,918774.10^{-5}$
SI30	$1,264588.10^{-5}$
C12	$4,096818.10^{-4}$

Finally, helium (table D.3) is located in the coolant tubes and in the annular region surrounding the fuel pellet.

$$N_{He} = \sum_{i \in \text{ F, C, K cells}} \left(\pi R_i^2 - \pi R_{i-1}^2\right) d_i \quad \text{and} \quad d_{He} = \frac{N_{He}}{S_{INSTANT}}$$

Table D.3 – Average density of helium in INSTANT

Isotope	Density $(.10^{24} atoms.cm^{-3})$
HE4	$1,372031.10^{-4}$

For the graphite (table D.4), the calculation was already done in the previous appendix to calculate the J-cell density. We have:

Table D.4 – Average density of graphite in INSTANT

Isotope	Density $(.10^{24} atoms.cm^{-3})$
GRAB	$6,901782.10^{-2}$

APPENDIX E

Comparison between P_1, P_3, P_5

In INSTANT, it is possible to choose how many angles are treated. This appendix shows a comparison between P_1, P_3 and P_5 to complete the explanations of section 3.2.2, page 58.

The first two tables concern P_1 and P_3 calculations. Table E.1 compares the K_{eff} and table E.2 the fission map parameters obtained in INSTANT. The variability is significant. On the contrary, the same K_{eff} are calculated between P_3 and P_5 (see table E.3). This demonstrates the convergence of the INSTANT model in P_3.

Table E.1 – Single block path: INSTANT calculations in P_1 and P_3, Eigenvalues

Nb. of groups in INSTANT	SB MOC DH295			
	P_1 scat 1		P_3 scat 1	
	Core K-eff	Deviation with MCNP (pcm)	Core K-eff	Deviation with MCNP (pcm)
295	1,25033	58	1,25010	40
26	1,25046	69	1,25023	50
23	1,24994	27	1,24970	8
12	1,24990	24	1,24968	6
10	1,25011	41	1,24987	22
9	1,24683	-221	1,24660	-240
6	1,24783	-141	1,24758	-162
4	1,20891	-3256	1,20891	-3256
2	1,21163	-3039	1,21162	-3039

Table E.2 – Single block path: INSTANT calculations in P_1 and P_3, fission map comparison

Nb. of groups in INST.	SB MOC DH295									
	P_1 scat 1					P_3 scat 1				
	Max	Min	Av. μ	Std σ	AbsD	Max	Min	Av. μ	Std σ	AbsD
295	2,06	-1,13	-0,15	0,94	0,82	1,83	-0,98	-0,14	0,81	0,69
26	2,02	-1,14	-0,15	0,98	0,88	1,79	-0,99	-0,13	0,85	0,74
23	2,05	-1,18	-0,15	1,03	0,93	1,83	-1,04	-0,14	0,90	0,80
13	1,49	-1,01	-0,11	0,87	0,81	1,27	-0,91	-0,10	0,73	0,68
10	1,53	-1,05	-0,11	0,90	0,84	1,30	-0,96	-0,10	0,76	0,71
9	0,24	-0,30	-0,007	0,18	0,16	0,35	-0,53	-0,009	0,48	0,25
6	0,98	-0,80	-0,07	0,61	0,58	0,76	-0,70	-0,06	0,29	0,45
4	11,2	-14,3	0,98	9,36	8,92	11,2	-14,5	0,99	9,36	9,01
2	10,7	-14,2	1,00	9,22	8,77	10,8	-14,4	1,01	9,22	8,86

Table E.3 shows the K_{eff} obtained with P_3 and P_5 calculations in INSTANT.

Table E.3 – Single block path: INSTANT calculations in P_3 and P_5, Eigenvalues

Nb. of groups in INSTANT	SB MOC DH295		
	P_3 scat 1 Core K-eff	P_5 scat 1 Core K-eff	Difference (pcm)
295	1,25010	1,25010	0
26	1,25023	1,25023	0
23	1,24970	1,24971	-0,8
12	1,24968	1,24967	0,8
10	1,24987	1,24987	0
9	1,24660	1,24660	0
6	1,24758	1,24758	0
4	1,20891	1,20891	0
2	1,21162	1,21161	0,8

APPENDIX F

Group structures

Tables F.1, F.2, F.3, F.4, F.5, F.6, F.7 and F.8 give the boundaries of the intermediate group-structures as they were determined by DRAGON. The boundaries are kept as close as possible to well-known group-structures. The lowest boundary is fixed at $1,100027.10^{-4}$ eV for every group structure.

All of these group-structures are based on boundaries given by publications, except the 10-group structure which has been built based on the 12-group structure by removing two boundaries which were located inside one larger interval of the 26-group structure. This enables condensation from 26 groups to 10 groups, while it was not possible to go from 26 groups to 12 groups with their existing boundaries.

Group structures with 26, 12, 6, 4, and 2 groups are given in the IAEA report [34] and have been used by the French CEA or the German Research Center of Jülich. Note that the 12-group structure is in fact based on the 13-group structure proposed in the report, but the last group was automatically removed by DRAGON because it was too low in energy. The group structure with 23 groups comes from the technical report [25] of Argonne National Laboratory. The 9-group structure has been determined by General Atomics and is cited in [35]. Once boundaries are given to DRAGON, the code finds the closest match with the internal boundaries of the 295-group structure. It explains the differences one may observe between the boundaries cited in those publications and the final boundaries chosen by the code and given in tables F.1 to F.8.

Table F.1 – 26-group structure, energy limits

26-group structure					
Group	1	2	3	4	5
E_{max} (eV)	1,9640E+07	6,7032E+06	3,3287E+06	5,7844E+05	9,4665E+04
Group	6	7	8	9	10
E_{max} (eV)	1,8585E+04	2,9962E+03	1,3436E+03	6,7729E+02	2,6830E+02
Group	11	12	13	14	15
E_{max} (eV)	1,2623E+02	5,9925E+01	2,7885E+01	1,3573E+01	7,9653E+00
Group	16	17	18	19	20
E_{max} (eV)	4,9585E+00	2,3301E+00	1,2509E+00	6,2500E-01	3,2501E-01
Group	21	22	23	24	25
E_{max} (eV)	1,9000E-01	1,2000E-01	7,6497E-02	4,7302E-02	1,4830E-02
Group	26				
E_{max} (eV)	7,1453E-03				

Table F.2 – 23-group structure, energy limits

23-group structure					
Group number	1	2	3	4	5
E_{max} (eV)	1,9640E+07	3,3287E+06	1,3369E+06	4,9400E+05	9,4665E+04
Group number	6	7	8	9	10
E_{max} (eV)	6,7379E+04	7,4658E+03	3,5357E+02	4,0000E+00	1,4440E+00
Group number	11	12	13	14	15
E_{max} (eV)	1,0920E+00	1,0350E+00	9,6396E-01	8,2004E-01	4,7502E-01
Group number	16	17	18	19	20
E_{max} (eV)	3,9000E-01	3,2501E-01	2,7999E-01	2,3119E-01	1,6190E-01
Group	21	22	23		
E_{max} (eV)	1,3800E-01	8,9797E-02	4,7302E-02		

Table F.3 – 12-group structure, energy limits

12-group structure					
Group	1	2	3	4	5
E_{max} (eV)	1,9640E+07	1,6507E+05	9,8249E+02	1,6562E+01	4,0000E+00
Group	6	7	8	9	10
E_{max} (eV)	2,0701E+00	1,2930E+00	7,2000E-01	6,2500E-01	3,9000E-01
Group	11	12			
E_{max} (eV)	3,0501E-01	1,0430E-01			

Table F.4 – 10-group structure, energy limits

10-group structure					
Group	1	2	3	4	5
$E_{\max}$ (eV)	1,9640E+07	1,6507E+05	9,8249E+02	1,6562E+01	4,0000E+00
Group	6	7	8	9	10
$E_{\max}$ (eV)	2,0701E+00	7,2000E-01	3,9000E-01	3,0501E-01	1,0430E-01

Table F.5 – 9-group structure, energy limits

9-group structure					
Group	1	2	3	4	5
$E_{\max}$ (eV)	1,9640E+07	1,6507E+05	9,0968E+02	1,7376E+01	3,8822E+00
Group	6	7	8	9	
$E_{\max}$ (eV)	2,3301E+00	1,2509E+00	8,2004E-01	1,1999E-01	

Table F.6 – 6-group structure, energy limits

6-group structure					
Group number	1	2	3	4	5
$E_{\max}$ (eV)	1,9640E+07	1,6507E+05	1,6562E+01	2,0701E+00	6,2500E-01
Group	6				
$E_{\max}$ (eV)	1,0430E-01				

Table F.7 – 4-group structure, energy limits

4-group structure				
Group number	1	2	3	4
$E_{\max}$ (eV)	1,9640E+07	9,4665E+04	2,7885E+01	1,7800E+00

Table F.8 – 2-group structure, energy limits

2-group structure		
Group number	1	2
$E_{\max}$ (eV)	1,9640E+07	2,3301E+00